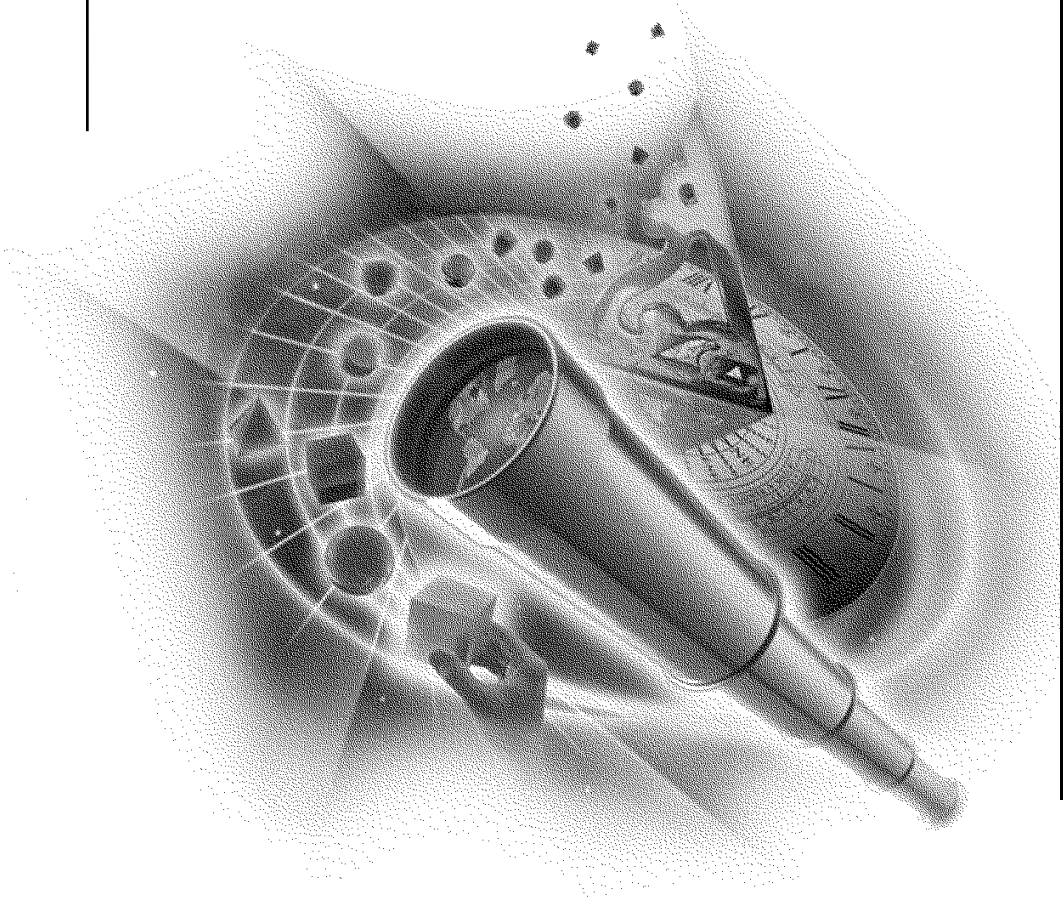
VERSION 3

ZENworks™ for Desktops

POLICY-BASED MANAGEMENT POWERED BY NDS® eDIRECTORY™

Administration



Novell®

Contents

About This Guide 13

1 Workstation Registration Files 17

Registration File Functions . . . 17
Registration File Locations . . . 17
WSREG.DLL Usage . . . 18

2 Reporting 21

The Effective Policies Report . . . 21
The Package Associations Report . . . 22

3 Windows 2000 Group Policies in Support Pack 1 23

Changes to Group Policies . . . 23
- Additive Group Policies . . . 23
- Revision Checking . . . 24
- Group Policy Caching . . . 24
- Persistent and Volatile Settings . . . 24

4 Setting Up Application Launcher/Explorer 27

Application Launcher/Explorer Overview . . . 27
- Application Launcher . . . 28
- Application Explorer . . . 29

Starting Application Launcher/Explorer . . . 31
- Manually Starting Application Launcher . . . 31
- Manually Starting Application Explorer . . . 31
- Automating Application Launcher Startup . . . 32
- Automating Application Explorer Startup . . . 32

Configuring Application Launcher/Explorer . . . 33
- Setting Configuration Options . . . 33
- Designating the Top of a Configuration Tree . . . 42

Using Application Launcher as the Windows Shell . . . 44
- Setting Up Application Launcher as the Shell on Windows 95/98 . . . 44
- Setting Up Application Launcher as the Shell on Windows NT/2000 . . . 45

Application Launcher/Explorer Command Line Switches . . . 45

5 Distributing Applications 55

Creating an Application's Installation Package . . . 56

Manually Creating an Installation Package . . . 56
Creating a snAppShot Installation Package . . . 56
Creating a Microsoft Windows Installer Package . . . 57
Creating an Application Object . . . 58
Creating a Simple Application Object . . . 59
Creating an Application Object With an .AOT/AXT File . . . 60
Creating an Application Object With an .MSI File . . . 60
Duplicating an Application Object . . . 61
Defining the Application's System Requirements . . . 62
Associating the Application Object with Users or Workstations . . . 63
Associating the Application Object with Users . . . 63
Associating the Application Object with Workstations . . . 65
Other Things You Can Do . . . 68

6 Uninstalling Applications 71

Enabling an Application to be Uninstalled . . . 71
Uninstalling Applications by Disassociating Users or Workstations . . . 74
Uninstalling Unused Applications . . . 75
Using Application Launcher/Explorer to Uninstall an Application . . . 76

7 Running in Disconnected Mode 79

Disconnectable Applications . . . 79
Workstation Cache . . . 80
Launch Cache . . . 80
Install Cache . . . 82
Configuring Application Launcher/Explorer to Start Automatically . . . 83
Configuring Applications as Disconnectable . . . 85
Forcing an Application to be Cached . . . 86
Changing the Default Cache Location . . . 87
Distributing Applications to Disconnected Workstations . . . 88
Enabling Users to Manage Applications . . . 90

8 Organizing Applications into Folders 93

Creating Application Folder Objects . . . 94
Adding Applications to Application Folders . . . 96
Using the Application Folder Object to Add Applications . . . 97
Using the Application Object to Add the Application to a Folder . . . 98
Adding Applications to Custom Folders . . . 99

9 Reporting on Application Management Events 101

Setting Up Database Reporting . . . 101
Installing the Sybase Database . . . 102
Creating a ZENworks Database Object . . . 102
Configuring the Database's JDBC and ODBC Information . . . 104
Granting ODBC Property Rights to Users . . . 106

Enabling the ZENworks Database Policy . . . 107
Configuring Application Objects to Use Database Reporting . . . 109
Setting Up SNMP Trap Reporting . . . 109
Enabling the SNMP Trap Targets Policy . . . 110
Configuring Application Objects to Use SNMP Trap Reporting . . . 112
Setting Up Log File Reporting . . . 112
Setting Up a Common Log File Location . . . 112
Configuring Application Objects to Use Log File Reporting . . . 113
Configuring Applications to Use Reporting . . . 113
Generating Database Reports . . . 115
Understanding Log File Reports . . . 115

10 Metering Software Licenses 119

Installing Novell Licensing Services . . . 119
Creating License Containers and Metered Certificates . . . 119
Configuring Applications to Use License Metering . . . 120

11 SnAppShot 123

Understanding SnAppShot . . . 123
Application Object Templates . . . 123
Application Source Files . . . 124
Application File Definition File . . . 124
Preparing a SnAppShot Workstation . . . 124
Creating an Installation Package . . . 125

12 Application Object Settings 127

Identification Tab . . . 127
Package Information Page (.MSI Application Object Only) . . . 127
Icon Page . . . 129
Description Page . . . 131
Folders Page . . . 132
Contacts Page . . . 134
Administrator Notes Page . . . 135
Distribution Options Tab . . . 136
Icons/Shortcuts Page (.AOT/.AXT Application Object Only) . . . 137
Registry Page . . . 139
Application Files Page (.AOT/.AXT Application Object Only) . . . 141
INI Settings Page . . . 145
Text Files Page (.AOT/.AXT Application Object Only) . . . 149
Distribution Scripts Page . . . 151
Pre-Install Schedule Page (.AOT/.AXT Application Object Only) . . . 153
Options Page . . . 156
Run Options Tab . . . 159
Application Page . . . 159
Environment Page . . . 162

Launch Scripts Page . . . 164
Environment Variables Page . . . 167
License/Metering Page . . . 168
Associations Tab . . . 169
Associations Page . . . 170
Availability Tab . . . 172
System Requirements Page . . . 173
Schedule Page . . . 188
Termination Page . . . 189
Common Tab . . . 193
Macros Page . . . 193
Drives/Ports Page . . . 197
File Rights Page . . . 200
Reporting Page . . . 202
Properties Page (.MSI Application Object Only) . . . 203
Transforms Page (.MSI Application Object Only) . . . 205
Imaging Page . . . 206
Sources Page . . . 209
Uninstall Page . . . 211
Fault Tolerance Tab . . . 214
Fault Tolerance Page . . . 215
Load Balancing Page . . . 217
Site List Page . . . 219

13 Macros 223

Application Object Macros . . . 224
Nesting Macros . . . 224
Defining an Application Object Macro . . . 225
Special Windows Macros . . . 226
Windows 95/98 and Windows NT/2000 Macros . . . 227
Windows 2000 Macros . . . 230
Login Script Macros . . . 231
Supported Login Script Macros . . . 231
Unsupported Login Script Macros . . . 233
NDS Attribute Macros . . . 234
Syntax . . . 234

Examples . . . 235
Environment Variable Macros . . . 236
Macro Precedence . . . 237
Application Object Properties Where Macros Can Be Used . . . 237

14 Application Launcher Tools 239

Export Application Object . . . 239
Show Inherited Applications . . . 239
Search and Replace . . . 240
Sync Distribution GUIDs . . . 241
Generate New GUIDs . . . 241
AOT/AXT File Tools . . . 241
Create Virtual CD . . . 242

15 Preparing Images 245

Creating a Workstation (Base) Image . . . 245
Creating an Add-On Image . . . 246
Customizing an Image . . . 247
Making an Image Available for Auto-Imaging . . . 247
Creating a Workstation Image Object . . . 248
Associating an Add-On Image with a Base Image . . . 249
Specifying to Use a Variant of an Image . . . 249

16 Multicasting Images 251

What Is Multicasting? . . . 251
Why Use Multicast? . . . 251
Multicast Procedure . . . 252

17 Imaging Utilities and Options 255

Image Explorer (IMGEXP.EXE) . . . 255
Imaging Agent (ZISWIN.EXE) . . . 256
Imaging Boot Disk Creator (ZIMGBOOT.EXE) . . . 258
Using ZIMGBOOT.EXE to Add Linux Drivers . . . 259
Imaging Bootup Parameters (SETTINGS.TXT) . . . 260
Imaging Bootup Languages (ZIMGLANG.INI) . . . 264
Imaging Engine (img Command) . . . 264
Help Mode . . . 265
Auto Mode . . . 266
Make Mode . . . 266
Restore Mode . . . 270
Dump Mode . . . 276
Partition Mode . . . 276
Information Mode . . . 278
Session (Multicast) Mode . . . 279
Imaging Server (IMGSERV.NLM or .DLL or .DLM) . . . 281

Imaging Server Log (ZIMGLOG.XML) . . . 283

18 Supported Ethernet Cards 287

Ethernet Cards for a Workstation . . . 287
Ethernet Cards for a Laptop Computer (PCMCIA) . . . 288

19 Understanding Remote Management Components 299

Understanding Remote Management Agents . . . 299
How the Remote Management Agent Uses Protocols . . . 300
Understanding Remote Management Security . . . 300
Understanding Remote Wake Up . . . 300
Understanding Remote Control . . . 301
Understanding Remote View . . . 301
Understanding Remote Execute . . . 301
Understanding Remote Diagnostics . . . 302
Understanding File Transfer . . . 303
Understanding Chat . . . 303
Understanding Audit Log . . . 303
Details of Events in the Audit Log . . . 304
Event Log Messages for Remote Management Sessions . . . 306
Monitoring Login and Logout events . . . 308
Specifying the Timeout Value for Establishing a Remote Control or Remote View Session . . . 309

20 Managing Remote Workstations 311

Using the Remote Management Agents . . . 311
Pinging the Remote Management Agent from the Management Console . . . 312
Shutting Down the Remote Management Agent . . . 312
Reloading the Remote Management Agent . . . 313
Uninstalling the Remote Management Agent using the Novell Application Launcher . . . 313
Managing a Remote Wake Up Session . . . 314
Managing a Remote Control Session . . . 315
Starting a Remote Control Session . . . 315
Enabling the Wallpaper on the Managed Workstation . . . 316
Controlling the Display of the Viewing Window . . . 316
Using the Viewing Window Accelerator Keys . . . 317
Using the Toolbar Buttons on the Viewing Window . . . 319
Using the Remote Management Icon . . . 321
Obtaining Information About Remote Management Sessions . . . 322
Stopping a Remote Control Session from the Managed Workstation . . . 324
Managing a Remote View Session . . . 324
Starting a Remote View Session . . . 325
Controlling the Display of the Viewing Window During a Remote View Session . . . 325
Stopping a Remote View Session from the Managed Workstation . . . 325
Running an Application on the Managed Workstation . . . 326
Viewing Diagnostic Information for a Managed Workstation . . . 326

Managing a File Transfer Session . . . 328
Starting a File Transfer Session . . . 329
Using File Transfer Window Controls . . . 329
Managing a Chat Session . . . 330
Chatting with a User at the Managed Workstation . . . 331
Editing the Messages in the Chat Window on the Management Console . . . 331
Viewing the Audit Log of Remote Management Sessions . . . 331

21 Diagnostic Information 333

Windows Memory Information . . . 334
Environment Information . . . 334
Event Log Information . . . 335
Device Drivers Information . . . 336
Services Information . . . 336
WIN32 Processes Information . . . 336
WIN32 Modules Information . . . 337
NetWare Connections Information . . . 337
Novell Client Information . . . 338
Network Protocols Information . . . 339
Name Space Providers Information . . . 340
Network Drives Information . . . 340
Network Open Files Information . . . 342
Print Capture Information . . . 343

22 Understanding Workstation Inventory 347

Understanding the Inventory Scanning Cycle . . . 347
Understanding Rolling Up Scan Data Across Servers . . . 349
Understanding the Inventory Scanner . . . 352
How the Scanners Collect Workstation Inventory Data . . . 353
Scanning Process Flowchart . . . 355
Summary of Files that the Scanner Processes . . . 356
Software Information Collected by the Scanners . . . 357
DMI-Compliant Scanners . . . 358
Hardware Data Collected by the Scanners . . . 360
Understanding the Sender-Receiver . . . 367
Understanding the Sender . . . 369
Understanding the Receiver . . . 370
Understanding the Compressed Scan Data File . . . 371
Quick Reference of Sender-Receiver . . . 371
Understanding the Selector . . . 375
Understanding the Storer . . . 377
Understanding the Service Manager . . . 379
List of Services . . . 379
Services on NetWare Servers . . . 383

Services on Windows NT/2000 Servers . . . 384
Inventory Upgrade Service . . . 385
Understanding the Effects of Workstation Inventory Installation . . . 386
List of Inventory Components on Servers . . . 390
Understanding ZENworks for Desktops 3 Inventory Attributes . . . 390
Understanding the ZENworks for Desktops 3 Inventory Database Schema . . . 416
Overview . . . 416
CIM Schema . . . 417
Inventory Database Schema in ZfD3 . . . 431

23 Setting Up Workstation Inventory 449

Configuring the Settings for the Inventory Service Object . . . 449
Configuring the Inventory Settings for the Workstation Object . . . 451
Configuring the Workstation Inventory Policy Settings . . . 452
Configuring the Roll-Up Policy Settings . . . 453
Scanning for IBM Computer Models . . . 454
Editing the IBMNAMES.INI File to Add the IBM Computer Models . . . 455
Customizing the Hardware Scanning Information of Jaz, Zip, and Floppy Drive Vendors . . . 456
Customizing Software Scanning of Workstations . . . 457
Adding New Applications for Scanning . . . 457
Format of the Custom Scan File . . . 458
Exporting the List of Application Files for Scanning . . . 459
Customizing the Software Scanning Information of Vendors and Products . . . 460
Changing the Role of the Inventory Service Object . . . 462
Changing the Role of the Root Server . . . 464
Changing the Role of the Root Server with Workstations . . . 466
Changing the Role of the Intermediate Server . . . 467
Changing the Role of the Intermediate Server with Database . . . 468
Changing the Role of the Intermediate Server with Database & Workstations . . . 470
Changing the Role of the Intermediate Server with Workstations . . . 471
Change the Role of the Leaf Server . . . 472
Changing the Role of the Leaf Server with Database . . . 474
Changing the Role of the Standalone Server . . . 475
Assigning Trustees to the Database Object . . . 477
Migrating ZENworks 2 Inventory Information . . . 477
Using ZENworks 2 Inventory Policies . . . 478
Migrating the Inventory Information from the ZENworks 2 Database . . . 479
Deleting the Inventory Information from ZENworks 2 database . . . 481
Scenarios: Migration of ZENworks 2 Inventory Information . . . 482
Setting up the Inventory Database . . . 485
Loading the Inventory Database as a Separate Oracle Instance . . . 486
Optimizing the Performance of the Oracle Database . . . 499
Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool) . . . 500

Using an Optimal Database Cache Size on the Inventory Database Server to Improve Performance . . . 502
Using the Inventory Database Service Configuration Tool for Windows NT/2000 . . . 503
Deleting the Inventory Information from the Inventory Database . . . 504
Synchronizing the Inventory Database with NDS (NDS-DB Sync Tool) . . . 504
Backing Up the Inventory Database . . . 505
Removing the ZENworks 1.*x* Inventory Attributes from Workstations . . . 510

24 Using Workstation Inventory 511

Viewing the Servers Deployed for Inventory . . . 511
Displaying Inventory Information . . . 511
Configuring the Inventory Database . . . 512
Viewing the Workstation Inventory Summary of a Managed Workstation . . . 513
Viewing the DMI Data in the Inventory Summary Information . . . 524
Viewing Inventory Information of Managed Workstations Formed by Querying the Database 525
Viewing the Minimal Inventory Information from an NDS Object . . . 527
Running Inventory Reports . . . 528
Exporting the Inventory Data to CSV Format . . . 534
Invoking the Data Export Tool . . . 534
Exporting the Inventory Data to a CSV File . . . 534
Forming the Query and Setting the Filter Conditions . . . 536
Loading Existing Configuration File . . . 537
Running the Data Export Program From the Server . . . 538

25 Troubleshooting Workstation Inventory with Status Logs 541

Viewing the Scan History of a Workstation . . . 541
Viewing the Roll-Up History of the Server . . . 542
Viewing the Status of Inventory Components on a Server . . . 543
Viewing the Status of the Last Scanning in the Workstation Scan Log . . . 544
Viewing the Roll-Up Log of Servers . . . 545
Exporting the Inventory Status Log Files . . . 546
Quick Overview of Status Logs and Scan Logs . . . 546
Viewing the Status Log in XML Format . . . 547

About This Guide

This Administration guide consists of comprehensive, conceptual information that you can use as you configure and use ZENworks™ for Desktops (ZfD) components in your production environment. The sections include:

- “Automatic Workstation Import” on page 15
- “Workstation Management” on page 19
- “Application Management” on page 25
- “Workstation Imaging” on page 243
- “Remote Management” on page 297
- “Workstation Inventory” on page 345

See the ZENworks for Desktops documentation Web site (http://www.novell.com/documentation/lg/zfd3/docui/index.html) for the most current ZfD administration information.

Automatic Workstation Import

This section provides information for ZENworks™ for Desktops (ZfD) Automatic Workstation Import and Removal:

- Chapter 1, “Workstation Registration Files,” on page 17.

For information on deploying Automatic Workstation Import or Removal, see Before the Deployment in *Deployment*.

1 Workstation Registration Files

Workstation registration in ZfD involves three executable files: WSREG32.EXE, WSREG.DLL, and WSREG32R.DLL.

The following sections provide information on these files:

- "Registration File Functions" on page 17
- "Registration File Locations" on page 17
- "WSREG.DLL Usage" on page 18

Registration File Functions

Following are the functions of the registration files:

WSREG32.EXE: If executed, calls WSREG.DLL.

WSREG.DLL: Performs workstation registration tasks.

WSREG32R.DLL: Contains strings used by WSREG.DLL. These strings are available for language translation.

Registration File Locations

Following are the locations of the registration files (by machine):

NetWare Server:

server_name\SYS\PUBLIC\WSREG32.EXE
server_name\SYS\PUBLIC\WSREG.DLL
server_name\SYS\PUBLIC\NLS\ENGLISH\WSREG32R.DLL

Windows 95/98 Client Workstation:

drive:*Windows_95/98_directory*\SYSTEM\WSREG32.EXE
drive:*Windows_95/98_directory*\SYSTEM\WSREG.DLL
drive:*Windows_95/98_directory*\SYSTEM\NLS\ENGLISH\WSREG32R.DLL

Windows NT/2000 Client Workstation:

drive:*Windows_NT/2000_directory*\SYSTEM32\WSREG32.EXE
drive:*Windows_NT/2000_directory*\SYSTEM32\WSREG.DLL
drive:*Windows_NT/2000_directory*\SYSTEM32\NLS\ENGLISH\WSREG32R.DLL

WSREG.DLL Usage

In addition to being called by executing WSREG32.EXE, WSREG.DLL can be called by Workstation Manager's Scheduler when when one of the following events occurs:

workstation startup
user login
user logout
registration service shutdown

The sequence of file calls when one of the above events is detected are:

Windows NT/2000: WM.EXE > WMRUNDLL.EXE > WSREG.DLL.

Windows 95/98: WM95.EXE > WM95SAST.EXE > WSREG.DLL.

II Workstation Management

This section provides information about the administration of ZENworks™ for Desktops (ZfD) Workstation Management, including:

- Chapter 2, “Reporting,” on page 21
- Chapter 3, “Windows 2000 Group Policies in Support Pack 1,” on page 23

For more information on Workstation Management, see Workstation Management in *Deployment*, which includes setup steps for all ZfD policies.

2 Reporting

ZENworks™ for Desktops (ZfD) provides predefined reports through ConsoleOne™ for effective policies and policy package associations.

You can run either report based on a selected container, and you can include its subcontainers.

Report results are automatically displayed in Notepad and are saved as text files in the *Windows_directory*\TEMP directory of the workstation where you are running ConsoleOne.

The following sections provide information on ZfD reporting:

- "The Effective Policies Report" on page 21
- "The Package Associations Report" on page 22

The Effective Policies Report

The effective policies report will show which policies are currently in effect for the listed objects.

The effective policies report provides the following information:

Version
Tree
Container
Object DN
Platform
Effective Policy DN

To run a report on the effective policies:

1. In ConsoleOne, click Tools > ZENworks Utilities > Report Policies and Packages.
2. In the Report From field, browse for a context for the report.
3. To include all subcontainers in that context, enable Include Subcontainers.
4. Click the Effective Policies Report radio button > click OK.

The report results are displayed in Notepad and automatically saved to:

drive:*Windows_directory*\TEMP\EFFECTIVEPOLICIES.TXT

on the workstation where you are running ConsoleOne.

The Package Associations Report

The package associations report will show which policy packages are associated with the listed containers, subcontainers, and objects.

The package associations report provides the following information:

Tree
Container
Package DN
Association

To run a report on policy package associations:

1. In ConsoleOne, click Tools > ZENworks Utilities > Report Policies and Packages.
2. In the Report From field, browse for a context for the report.
3. To include all subcontainers in that context, enable Include Subcontainers.
4. Click the Package Association Report radio button > click OK.

The report results are displayed in Notepad and automatically saved to:

drive:*Windows_directory*\TEMP\PACKAGEASSOCIATIONS.TXT

on the workstation where you are running ConsoleOne.

3 Windows 2000 Group Policies in Support Pack 1

NOTE: Support Pack 1

All of the information in the following sections pertains to the new properties of Group policies starting with Windows* 2000 Group policies.

Changes to Group Policies

Group policies have changed significantly since the ZENworks™ for Desktops (ZfD) version 3 initial release. Group policies are now additive, they check for revisions, they cache already-processed policies, and they use persistent or volatile settings.

IMPORTANT: The Group policy path (Network Location of Existing/New Group Policies input field) should be a UNC path. Paths with mapped drives may not work because Group policies can run before the login script is processed. Future revisions of the Group policy editor in ConsoleOne™ will automatically convert paths to UNC.

You can find more information in the following sections:

- "Additive Group Policies" on page 23
- "Revision Checking" on page 24
- "Group Policy Caching" on page 24
- "Persistent and Volatile Settings" on page 24

Additive Group Policies

Group policies are now additive. This means that settings from multiple Group policies are cumulatively effective, rather than individually. Settings from multiple Group policies can affect users and workstations. Policies start with the local Group policy settings and are applied in reverse of the policy search order. This means that a setting in a policy applied first has lowest priority and its value is overwritten by any other policy with the same setting.

Security settings are not additive; they are set by the last effective policy.

Revision Checking

Group policies now track the revision of the policies in effect. As long as the list of effective policies and their revisions remains the same, Group policies are not processed, but use the cached Group policy.

NOTE: Each time the Edit Policies button is clicked, the revision of a Group policy changes, causing the policies to be reprocessed.

Group Policy Caching

The last-processed Group policy is cached locally. This helps reduce network traffic by processing Group policies only if necessary. If UserA logs in on a new machine, his or her effective Group policies are processed and then cached.

If UserA logs out and UserB logs in, and if UserB has the same effective Group policies as UserA, the locally-cached Group policy is restored instead of reprocessing Group policies. If the list of effective policies is different or if the revision is changed on any policy, the Group policies are reprocessed.

Persistent and Volatile Settings

The administrator determines if Group policies are persistent or volatile. The persistent setting indicates that when the Group Polices are set, they remain set—even if a user happens to log in only to a workstation and not to the network.

The volatile setting indicates that the original local Group policy settings will be restored when:

- The user logs out (user settings are restored)
- The system shuts down (workstation settings are restored)

III Application Management

The following sections provide information on ZENworks™ for Desktops (ZfD) Application Management tasks you may need to perform and features you might want to use.

- Chapter 4, “Setting Up Application Launcher/Explorer,” on page 27
- Chapter 5, “Distributing Applications,” on page 55
- Chapter 6, “Uninstalling Applications,” on page 69
- Chapter 7, “Running in Disconnected Mode,” on page 75
- Chapter 8, “Organizing Applications into Folders,” on page 89
- Chapter 9, “Reporting on Application Management Events,” on page 97
- Chapter 10, “Metering Software Licenses,” on page 115
- Chapter 11, “SnAppShot,” on page 119
- Chapter 12, “Application Object Settings,” on page 123
- Chapter 13, “Macros,” on page 217
- Chapter 14, “Application Launcher Tools,” on page 233

4 Setting Up Application Launcher/ Explorer

Novell® Application Launcher™ and Application Explorer (Application Launcher/Explorer) run on users' workstations to deliver the applications you've configured in NDS®. The following sections provide information to help you understand and set up Application Launcher/Explorer.

- "Application Launcher/Explorer Overview" on page 27
- "Starting Application Launcher/Explorer" on page 31
- "Configuring Application Launcher/Explorer" on page 33
- "Using Application Launcher as the Windows Shell" on page 44
- "Application Launcher/Explorer Command Line Switches" on page 45

Application Launcher/Explorer Overview

Application Launcher and Application Explorer are 32-bit applications that support Windows* 95, Windows 98, Windows NT*, and Windows 2000. When a user starts Application Launcher/Explorer, Application Launcher/Explorer reads NDS to provide access to the applications the user has been given rights to.

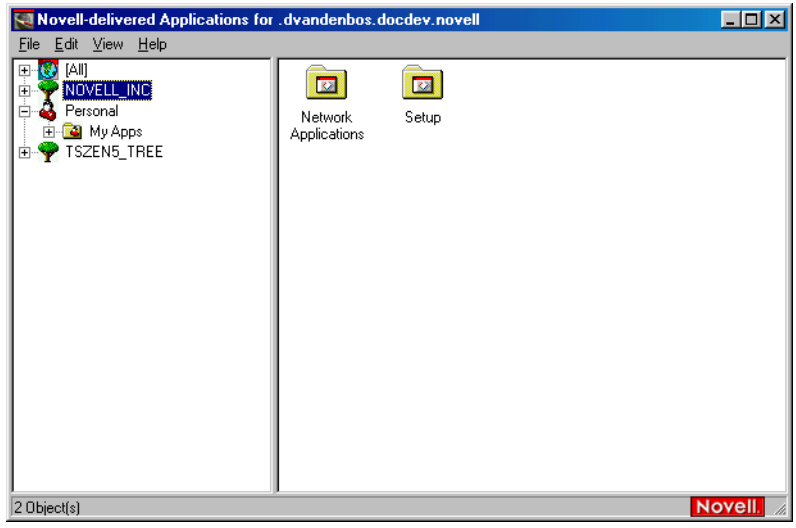
NOTE: If the user is not connected to NDS, Application Launcher/Explorer reads the user's local cache to see what applications the user has access to while disconnected from NDS.

While Application Launcher and Application Explorer perform the same basic functions and both run on Windows 95/98 and Windows NT/2000, Application Launcher is designed to enable greater administrative control of the user's desktop while Application Explorer is designed to enhance and extend the desktop. These differences are explained in greater detail in the following sections.

- "Application Launcher" on page 28
- "Application Explorer" on page 29

Application Launcher

Application Launcher, shown below, is a standalone window that can be accessed from the workstation's desktop.



NOTE: You can replace the Windows desktop with the Application Launcher windows. For instructions, see "Using Application Launcher as the Windows Shell" on page 44.

The Application Launcher window is divided into two panes, much like Windows Explorer. In the left pane, Application Launcher displays the following:

- [All] folder: Contains all applications that have been distributed to the user. This is an administrator-controlled feature. By default, it is enabled, which means the folder appears. You can disable the feature if desired.
- NDS trees: Each tree contains the applications, located within the tree, that have been distributed to the user or workstation. Application Launcher displays only the trees to which the user is authenticated.
- Personal folder: Provides a location for the user to create personal folders for organizing applications. This is an administrator-controlled feature. By default, it is disabled, which means the folder does not appear.

When the user selects a tree or folder in the left pane, the right pane displays the items (folders or applications) that are contained within the tree or folder.

Using the Application Launcher window, the user can do the following:

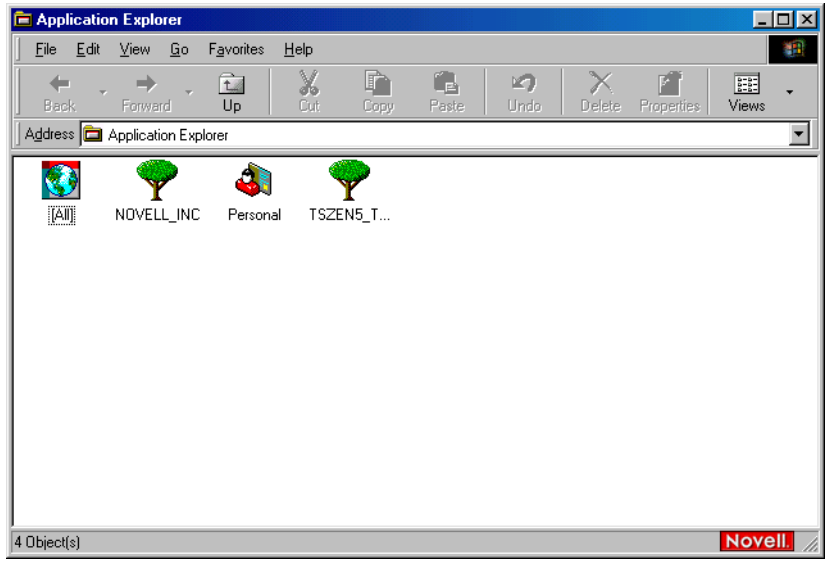
- Run an application by double-clicking the application's icon in the right pane. Depending on the application and how it is configured in NDS, before running the application Application Launcher may install files to the workstation, map drives, or change workstation configuration files or settings.
- View the properties of an application. The properties include a description of the application, information about people to contact for help with the application, the times when the application is available for use, and the workstation requirements established for the application.
- Log in to NDS (or to another NDS tree).
- Disconnect Application Launcher from NDS so that the user can work offline. Disconnecting Application Launcher from NDS does not log the user out of NDS.

As administrator, you control the availability of certain features that enable users to do the following:

- Uninstall an application.
- Cache an application's installation files to the user's workstation so that the user can install or verify (fix) the application while disconnected from NDS.

Application Explorer

Application Explorer, shown below, is also a standalone window that can be accessed from the Windows desktop.



In addition, with Application Explorer, you can configure Application objects to display on the Windows desktop, the Start menu, the system tray, and the Quick Launch tool bar.

The Application Explorer window displays the following:

- [All] folder: Contains all applications that have been distributed to the user. This is an administrator-controlled feature. By default, it is enabled, which means the folder appears. You can disable the feature if desired.
- NDS trees: Each tree contains the applications, located within the tree, that have been distributed to the user or workstation. Application Launcher displays only the trees to which the user is authenticated.
- Personal folder: Provides a location for the user to create personal folders for organizing applications. This is an administrator-controlled feature. By default, it is disabled, which means the folder does not appear.

Using the Application Explorer window, the user can do the following:

- Run an application by double-clicking the application's icon. Depending on the application and how it is configured in NDS, before running the application Application Explorer may install files to the workstation, map drives, or change workstation configuration files or settings.
- View the properties of an application. The properties include a description of the application, information about people to contact for

help with the application, the times when the application is available for use, and the workstation requirements established for the application.

- Disconnect Application Explorer from NDS so that the user can work offline. Disconnecting Application Launcher from NDS does not log the user out of NDS.

Starting Application Launcher/Explorer

The following sections explain how to manually start Application Launcher and Application Explorer and how to automate the starting of the applications:

- "Manually Starting Application Launcher" on page 31
- "Manually Starting Application Explorer" on page 31
- "Automating Application Launcher Startup" on page 32
- "Automating Application Explorer Startup" on page 32

Manually Starting Application Launcher

To manually start Application Launcher on a workstation:

1. Make sure that the Novell Client™ for Windows 95/98 (version 3.3 or later) or the Novell Client for Windows NT/2000 (version 4.8 or later) is installed on the workstation.
2. Run NAL.EXE from the SYS:\PUBLIC directory on a server where you installed Application Management.

NAL.EXE copies Application Launcher files to the workstation and starts Application Launcher.

Manually Starting Application Explorer

To manually start Application Explorer on a workstation:

1. Make sure that the Novell Client for Windows 95/98 (version 3.3 or later) or the Novell Client for Windows NT/2000 (version 4.8 or later) is installed on the workstation.
2. Run NALEXPLD.EXE from the SYS:\PUBLIC directory on a server where you installed Application Management.

NALEXPLD.EXE copies Application Explorer files to the workstation and starts Application Explorer.

Automating Application Launcher Startup

To automatically start Application Launcher each time a user logs in to NDS:

1. Make sure that the Novell Client for Windows 95/98 (version 3.3 or later) or the Novell Client for Windows NT/2000 (version 4.8 or later) is installed on the workstation.
2. Enter the following line in the user's login script:

   @*servername*\sys\public\nal.exe

   where *servername* is the actual name of your network server.

   You can enter this line in a container, profile, or user login script. If the login script also contains an entry for the Automatic Client Upgrade (ACU), make sure the NAL.EXE entry is listed before the ACU entry.

   IMPORTANT: You can also add Application Launcher to the Windows Startup folder. This enables Application Launcher to start when the workstation is disconnected from NDS. You configure this option in ConsoleOne™ and can have it apply to a single user, a group of users, or all users in a container. To do so, right-click a User, Group, or container object > click Properties > click the Application Launcher tab. On the Application Launcher page, click Edit > click the User tab > scroll to select the Auto-Start NAL When Disconnected option > set the option to Yes.

Automating Application Explorer Startup

To automatically start Application Explorer each time the user logs in to NDS:

1. Make sure that the Novell Client for Windows 95/98 (version 3.3 or later) or the Novell Client for Windows NT/2000 (version 4.8 or later) is installed on the workstation.
2. Enter the following line in the user's login script:

   @*servername*\sys\public\nalexpld.exe

   where *servername* is the actual name of your network server.

   You can enter this line in a container, profile, or user login script. If the login script also contains an entry for the Automatic Client Upgrade (ACU), make sure the NALEXPLD.EXE entry is listed before the ACU entry.

IMPORTANT: You can also add Application Explorer to the Windows Startup folder. This enables Application Explorer to start when the workstation is disconnected from NDS. You configure this option in ConsoleOne and can have it apply to a single user, a group of users, or all users in a container. To do so, right-click a User, Group, or container object > click Properties > click the Application Launcher tab. On the Application Launcher page, click Edit > click the User tab > scroll to select the Auto-Start NAL When Disconnected option > set the option to Yes.

Configuring Application Launcher/Explorer

Application Launcher/Explorer includes configuration options you can use to determine how Application Launcher/Explorer works and the functionality that Application Launcher/Explorer exposes to users.

You use ConsoleOne to set configuration options at the container level, user level, or workstation level. When Application Launcher/Explorer searches the NDS tree for a user's (or workstation's) Application Launcher/Explorer settings, it starts with the User (or Workstation) object. If the option has not been set for the User (or Workstation) object, Application Launcher/Explorer looks at the object's parent container to see if the option has been set at that level. If not, Application Launcher/Explorer continues up the NDS tree searching for the option's setting until it reaches a container object that has been designated as the top of the configuration tree. If it doesn't find any settings, then the setting is considered "unset" and the default setting is applied. Every configuration option has a default setting.

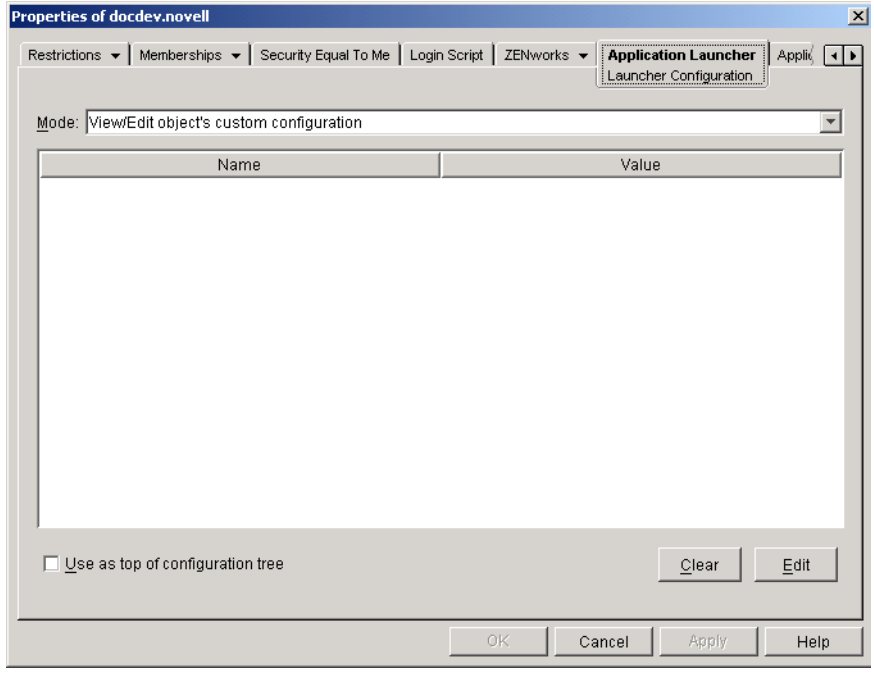
The following sections explain how to set configuration options and how to designate an object as the top of a configuration tree.

- "Setting Configuration Options" on page 33
- "Designating the Top of a Configuration Tree" on page 42

Setting Configuration Options

To set configuration options:

1. In ConsoleOne, right-click the User, Workstation, or container object to which you want to apply new settings > click Properties.
2. Click the Application Launcher tab to display the Launcher Configuration page.



The Launcher Configuration page provides three modes you can use to view the configuration settings for the current object. By default, the View/Edit Object's Custom Configuration mode is selected.

3 Make sure the View/Edit Object's Custom Configuration mode is selected. This is the mode that lets you set configuration options for the object. All three modes are described below.

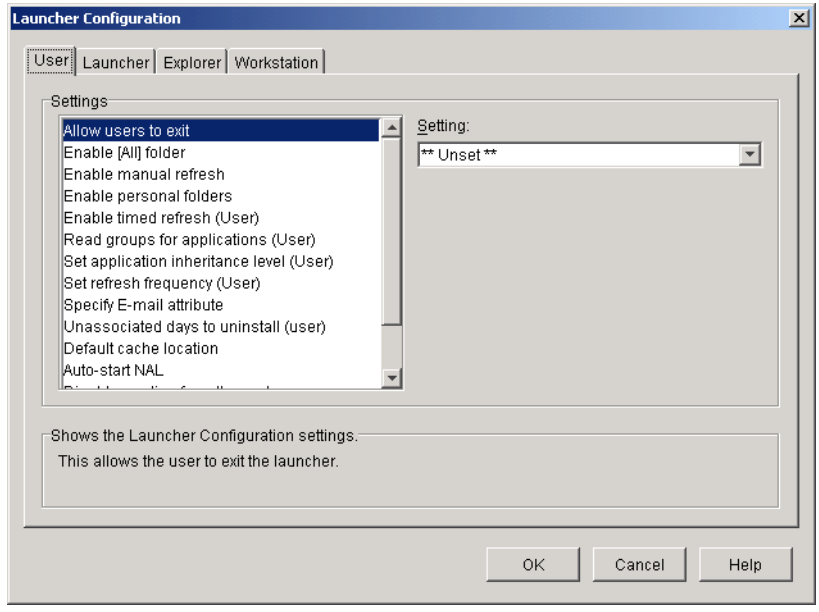
View/Edit Object's Custom Configuration (Default Mode): In this mode, the window lists all configuration options that have been set, or "customized," and lists the settings values. Options that are unset, meaning they receive their settings from their parent container (or higher), are not displayed.

This is the mode you must use to set a configuration option.

View Object's Effective Settings: In this mode, the window lists all configuration options with their effective settings. Effective settings may be from custom settings applied directly to the object, from settings inherited from a parent container, or from settings determined by the default settings values. This mode is useful when you want to see all settings that are being applied to the object.

View Configuration Tree: In this mode, the window displays the portion of the NDS tree that is being used to determine the configuration settings for the option. Only configuration options that are using a custom setting (either from the current object or from its parent container) are displayed. This mode is useful when you want to see what object a setting is being inherited from.

4 Click Edit to display the Launcher Configuration dialog box.



The Launcher Configuration dialog box has three or four tabs, depending on the type of object you are setting configuration options for: User, Launcher, Explorer, and Workstation.

User: Includes options that apply to both Application Launcher and Application Explorer.

Launcher: Includes options that apply to Application Launcher only.

Explorer: Includes options that apply to Application Explorer only.

Workstation: Includes options that apply to both Application Launcher and Application Explorer, but only when the user's workstation has been imported into NDS as a Workstation object and Workstation Manager is running on the workstation.

5 Click User > set the following options:

Allow Users to Exit: Specify whether or not to allow the user to exit Application Launcher/Explorer. If you allow the user to exit, you should let him or her know how to start Application Launcher/Explorer again.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Enable [All] Folder: Specify whether or not the user can see the [All] applications folder in Application Launcher/Explorer. The [All] applications folder displays all applications the user has access to, regardless of the application's NDS tree or folder.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Enable Manual Refresh: Specify whether or not the user can refresh Application Launcher/Explorer to distributed any changes since the last refresh. The Enable Manual Refresh and Enable Timed Refresh options are not connected in any way except that they both control refresh. One option does not need to be selected for the other to work.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Enable Personal Folders: Specify whether or not the user can create Application Launcher/Explorer folders and move the icons in them as he or she sees fit. Only Application object icons (or associated application icons or shortcuts) can be moved into personal folders. Use caution when offering the option if you want to exert more control and thus reduce support calls.

The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

Enable Timed Refresh: Specify whether or not Application Launcher/Explorer will automatically refresh the application so the user doesn't have to manually refresh icons.

The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

Read Group Objects for Applications: Specify whether or not you want Application Launcher/Explorer to read Group objects to see if the user has been associated with any applications through membership in a group. Although groups are a convenient way of indirectly associating the user with applications, requiring Application Launcher/Explorer to read Group objects can also decrease performance.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Set Application Inheritance Level: Specify how many parent containers Application Launcher/Explorer should search for applications associated with the user.

The setting values are -1, 1 through 999, and Unset. Entering -1 instructs Application Launcher/Explorer to search to the root of the NDS tree. The default value (1) is used if you select Unset and no parent container includes a customized setting.

Set Refresh Frequency: Specify how often you want Application Launcher/Explorer to search NDS for new or changed Application objects associated with the user. A short timed refresh interval is very useful in situations where you want changes to refresh quickly. However, a short timed refresh interval can cause higher network traffic.

The setting values are 0 through 999,999,999 (seconds) and Unset. The default value (3600) is used if you select Unset and no parent container includes a customized setting.

Specify E-Mail Attribute: Specify the NDS attribute you want to use to display e-mail names on the Help Contacts tab of the Application object's Properties dialog box. If the user has problems with applications, he or she can contact people by e-mail to get help. The e-mail name that appears is pulled from the NDS attribute you specify here. Users must have the NDS rights required to read the selected attribute for all users defined as contacts.

The setting values include all the NDS attributes and Unset. The default value (Mailbox ID) is used if you select Unset and no parent container includes a customized setting

Unassociated Days to Uninstall: Specify the number of days after which you want Application Launcher/Explorer to uninstall an application when the user has been unassociated with the application.

The setting values are 0 through 730 and Unset. Specify 0 if you want the application to be uninstalled as soon as the user is no longer associated with it. The default value (0) is used if you select Unset and no parent container includes a customized setting.

NOTE: This option requires you to have enabled the application to be uninstalled. For information, see Chapter 6, "Uninstalling Applications," on page 69.

Default Cache Location: Select the drive where Application Launcher/Explorer should create the cache directory. The cache directory contains 1) the information required to uninstall the applications and 2) the files needed to install or repair the application when the workstation is disconnected from NDS (for example, a laptop).

The setting values are drives A through Z and the Windows Drive. The default is the Windows Drive.

Auto-Start NAL: Typically, Application Launcher/Explorer is started via a login script when a user logs in to NDS. Specify whether or not you want Application Launcher/Explorer to be included in the user's Startup folder so that it will be automatically started even when the user does not log in to NDS. This option will not cause Application Launcher/Explorer to run twice (from the login script and the Startup folder) when the user logs in to NDS.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Disable Reading from the Cache: Specify whether or not you want to disable Application Launcher/Explorer's ability to read information from the cache directory.

The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

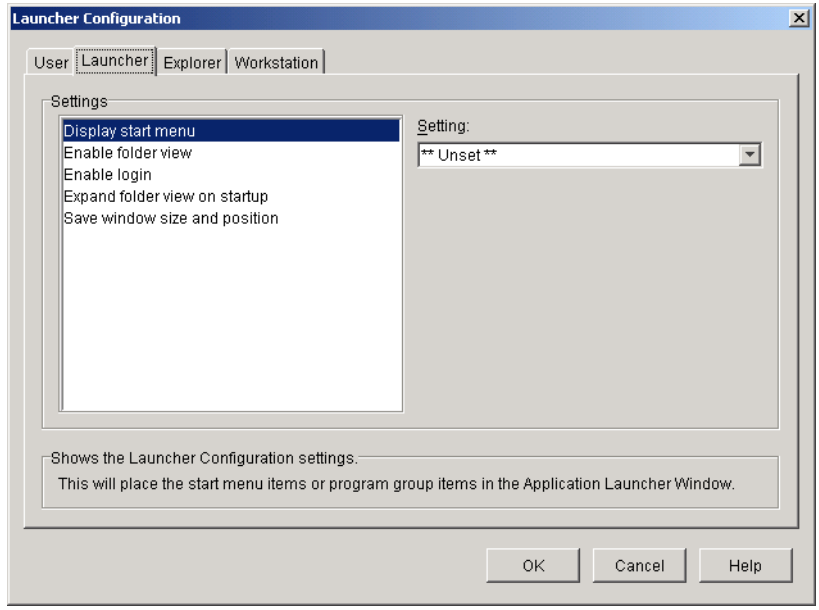
Always Evaluate Referrals: The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

The setting values are Yes, No, and Unset. The default value (No) forces all NDS calls to the same replica and is used if you select Unset and no parent container includes a customized setting.

Enable Manage Applications Dialog: Specify whether or not you want to enable Application Launcher/Explorer's Manage Applications dialog box. This dialog box lets the user decide which disconnectable applications to include in the local cache and which cached applications to install or uninstall.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

6 Click Launcher > set the following options:



Display Start Menu: Specify if and when you want Application Launcher to display the current Windows 95/98 or Windows NT/2000 Start menu organization (including Programs and above) in the Application Launcher window.

The setting values are Show When Not Authenticated, Show When Authenticated, Show Always, Never Show, and Unset. The default value (Show When Not Authenticated) is used if you select Unset and no parent container includes a customized setting.

Enable Folder View: Specify whether or not Application Launcher displays folders in the Application Launcher window.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Enable Log In: Specify whether or not you want to activate the Log In option found on the File menu in the Application Launcher window so the user can run the Novell Client to log in to NDS. Before you enable the Log In option, ensure that Application Launcher can find the login program (LOGINW32.EXE) on the user's workstation. Put the login program in the startup directory, working directory, or path.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

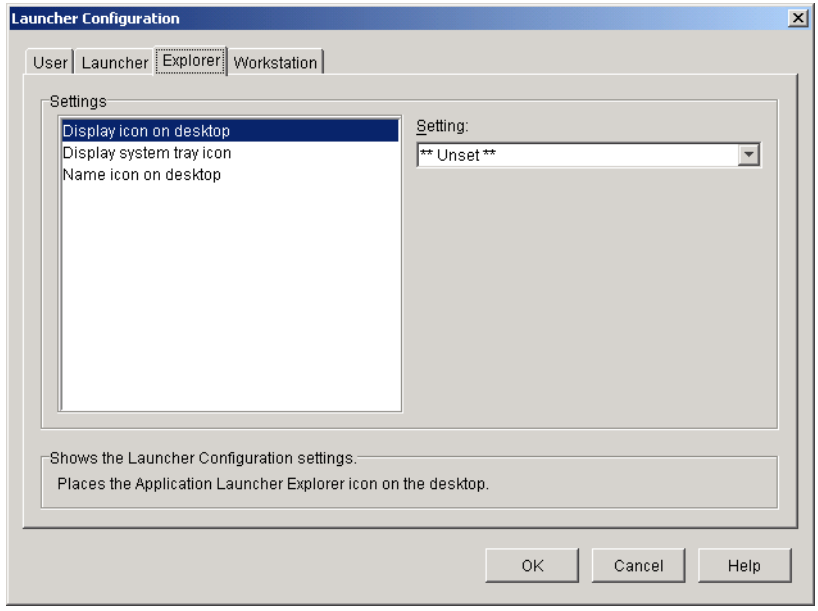
Expand Folder View on Startup: Specify whether or not you want Application Launcher to expand the entire folder tree when it starts.

The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

Save Window Size and Position: Specify whether or not you want Application Launcher to save its window size and position settings on a local drive.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

7 Click Explorer > set the following options:



Display Icon on Desktop: Specify whether or not you want the Application Explorer icon displayed on the user's desktop. The Application Explorer icon lets users open the Application Explorer window to access the applications that have been distributed to them.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Display System Tray Icon: Specify whether or not you want the Application Explorer icon displayed in the Windows system tray. The

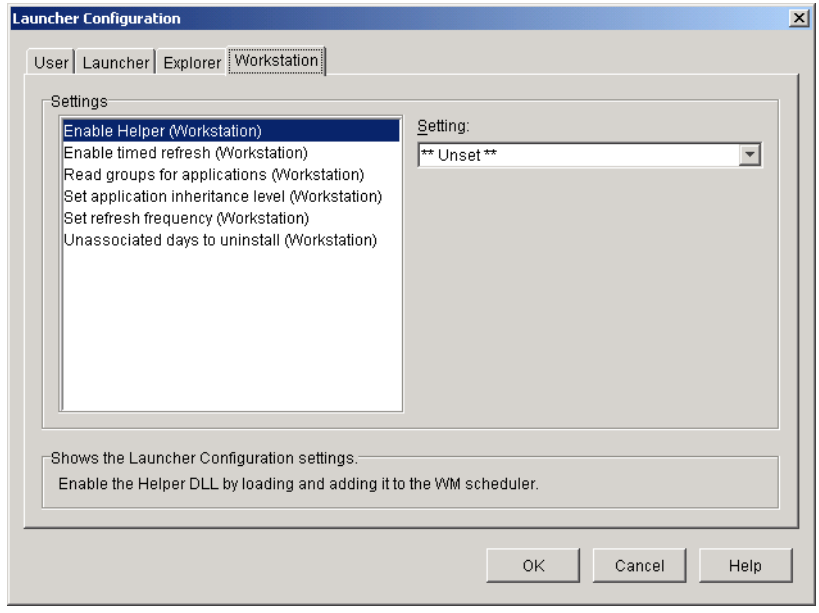
Application Explorer icon lets users open the Application Explorer window to access the applications that have been distributed to them.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Name Icon on Desktop: Use this setting to change the name of Application Explorer's desktop icon. The default name is Application Explorer.

The setting values are Custom and Unset. The default value (Application Explorer) is used if you select Unset and no parent container includes a customized setting. To use a different name, select Custom and enter the icon title.

8 Click Workstation > set the following options:



Enable Helper: Specify whether or not you want to have the Workstation Helper DLL installed and run by the Workstation Management Scheduler.

The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Enable Timed Refresh: Specify whether or not you want to automatically refresh application icons.

The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

Read Group Objects for Applications: Specify whether or not you want to read Group objects to see if the workstation has been associated with any applications through membership in a group. Although groups are a convenient way of indirectly associating workstations with applications, reading Group objects can also decrease performance.

The setting values are Yes, No, and Unset. The default value (No) is used if you select Unset and no parent container includes a customized setting.

Set Application Inheritance Level: Specify how many parent containers should search for applications associated with the workstation.

The setting values are 1 through 999 and Unset. The default value (1) is used if you select Unset and no parent container includes a customized setting.

Set Refresh Frequency: Specify how often you want to search NDS for new or changed Application objects associated with the workstation. A short timed refresh interval is very useful in situations where you want changes to refresh quickly. However, a short timed refresh interval can cause higher network traffic.

The setting values are 0 through 999,999,999 (seconds) and Unset. The default value (3600) is used if you select Unset and no parent container includes a customized setting.

9 Click OK to return to the Launcher Configuration page > click OK to save the changes.

Designating the Top of a Configuration Tree

When Application Launcher/Explorer searches the NDS tree for a user's (or workstation's) Application Launcher/Explorer settings, it starts with the User (or Workstation) object. If the option has not been set for the User (or Workstation) object, Application Launcher/Explorer looks at the object's parent container to see if the option has been set at that level. If not, Application Launcher/Explorer continues up the NDS tree searching for the option's setting until it reaches an object that has been designated as the top of the tree.

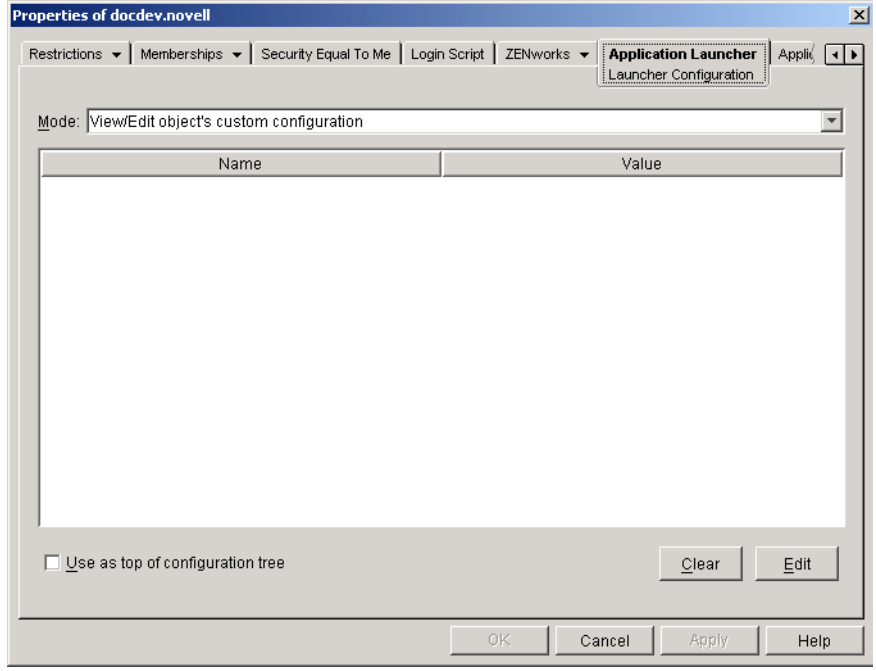
You can designate User, Workstation, or container objects as the top of a configuration tree. For example, if you want Application Launcher/Explorer to read only the user's object for its configuration settings, you would

designate the User object as the top of the configuration tree. Or, if you want Application Launcher/Explorer to read the user's object and its parent container object for its configuration settings, you would designate the parent container object as the top of the configuration tree.

Not all users and workstations must use the same top for their configuration tree. For example, for one user you may want Application Launcher/Explorer to read the user's configuration settings only, while you want to use the parent container as the top of the configuration tree for all other user in the container. In this case, you would specify the User object as the top of the configuration tree for the one user, and specify the parent container as the top of the configuration tree for the other users.

To designate an object as the top of the configuration tree:

1. In ConsoleOne, right-click the User, Workstation, or container object you want to designate as the top of a configuration tree > click Properties.
2. Click the Application Launcher tab to display the Launcher Configuration page.



3 Check the Use as Top of Configuration Tree box to designate this object as the top of a configuration tree.

4 Click OK.

Using Application Launcher as the Windows Shell

You can use Application Launcher in place of Windows Explorer (the default Windows shell) to further restrict user access to applications on Windows 95/98 or Windows NT/2000 workstations.

- "Setting Up Application Launcher as the Shell on Windows 95/98" on page 44
- "Setting Up Application Launcher as the Shell on Windows NT/2000" on page 45

Setting Up Application Launcher as the Shell on Windows 95/98

1 Copy the following files to the locations specified.

File	Copy From	Copy To
NALWIN32.EXE	*Z*:\PUBLIC	C:\NOVELL\CLIENT32
NALRES32.DLL	*Z*:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NALBMP32.DLL	*Z*:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NALEXP32.HLP	*Z*:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NALEXP32.CNT	*Z*:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NWAPP32.DLL	*Z*:\PUBLIC	C:\NOVELL\CLIENT32

2 Open the workstation's SYSTEM.INI file (typically, C:\WINDOWS\SYSTEM.INI) with a text editor.

3 Replace the SHELL=EXPLORER.EXE line with the following line:

```
SHELL=C:\NOVELL\CLIENT32\NALWIN32.EXE
```

4 Save and close the SYSTEM.INI file.

5 Restart Windows.

Setting Up Application Launcher as the Shell on Windows NT/2000

1 Copy the following files to the locations specified.

File	Copy From	Copy To
NALWIN32.EXE	*Z*:\PUBLIC	C:\WINNT\SYSTEM32
NALRES32.DLL	*Z*:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NALBMP32.DLL	*Z*:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NALEXP32.HLP	*Z*:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NALEXP32.CNT	*Z*:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NWAPP32.DLL	*Z*:\PUBLIC	C:\WINNT\SYSTEM32

2 Run REGEDIT.EXE and locate the following setting:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon
```

3 Modify the SHELL value as follows:

```
SHELL=C:\WINNT\SYSTEM32\NALWIN32.EXE
```

4 Close REGEDIT.EXE.

5 Restart Windows.

Application Launcher/Explorer Command Line Switches

The following command line switches can be used when running Application Launcher (NAL.EXE) or Application Explorer (NALEXPLD.EXE).

Switch	Description
SYNTAX: : APPLICATION LAUNCHER EXAMPLE: nal: APPLICATION EXPLORER EXAMPLE: nalexpld:	This switch is valid for Application Launcher only. The /ns switch provides the same functionality for Application Explorer. Skips the initial splash screen.
SYNTAX: /a=".*application_object*" or /a="*tree*:.*application_object*" APPLICATION LAUNCHER EXAMPLES: nal /a=".snapshot.apps" or nal /a="novell:.snapshot.apps" APPLICATION EXPLORER EXAMPLES: nalexpld /a=".snapshot.apps" or nalexpld /a="novell:.snapshot.apps"	Distributes and launches the specified application object. *Application_object* is the full object name of the application to run. *Tree* is the name of the tree in which the object exists. The default tree is assumed if no tree name is specified. **NOTE:** The quotation marks are required only if a space is used in the tree name or Application object name.

Switch	Description
SYNTAX: /c="*window_title*" APPLICATION LAUNCHER EXAMPLE: nal /c="Marketing Applications for %CN%" APPLICATION EXPLORER EXAMPLE: nalexpld /c="Marketing Applications for %CN%"	Uses the specified text as the title for the Application Launcher/Explorer window. The default window title is "Novell-delivered Applications for *User*," where *User* is the distinguished name of the user. The text specified can contain macros (variables) to display NDS information. The example, /c="Marketing Applications for %CN%" will display "Marketing Applications for JSmith" for a user logged in as JSmith. For information about macros, see Chapter 13, “Macros,” on page 217.
SYNTAX: /dm APPLICATION LAUNCHER EXAMPLE: nal /dm APPLICATION EXPLORER EXAMPLE: nalexpld /dm	Displays the installation progress of Microsoft* Windows Installer if the Windows Installer is not already installed on the workstation.
SYNTAX: /f APPLICATION LAUNCHER EXAMPLE: nal /f APPLICATION EXPLORER EXAMPLE: nalexpld /f	Instructs Application Launcher/ Explorer not to read NDS to get the list of applications that have been associated with the logged-in user or the workstation. This is useful when using the /a, /v, or /l switches to simply distribute, verify, or uninstall a specific application. This switch also requires you to use the /h switch.

Switch	Description
SYNTAX: /h APPLICATION LAUNCHER EXAMPLE: nal /h APPLICATION EXPLORER EXAMPLE: nalexpld /h	Instructs Application Launcher/ Explorer to hide its window from view when loaded. This switch is required when using the /f switch and is useful with the /i, /l, and /v switches.
SYNTAX: /i APPLICATION LAUNCHER EXAMPLE: nal /i APPLICATION EXPLORER EXAMPLE: nalexpld /i	Installs or updates the Application Launcher/Explorer files on the local hard drive, but does not start Application Launcher/Explorer. This switch can be used on a secured NT workstation on which the SYSTEM32 directory is set to "read only" access for the user.

Switch	Description
SYNTAX: /l="*.application_object*" or /l="*tree*:*.application_object*" APPLICATION LAUNCHER EXAMPLES: nal /l=".snapshot.apps" or nal /l="novell:.snapshot.apps" APPLICATION EXPLORER EXAMPLES: nalexpld /l=".snapshot.apps" or nalexpld /l="novell:.snapshot.apps"	Uninstalls the specified application object. *Application_object* is the full object name of the application to run. *Tree* is the name of the tree in which the object exists. The default tree is assumed if no tree name is specified. **NOTE:** The quotation marks are required only if a space is used in the tree name or Application object name.
SYNTAX: /max APPLICATION LAUNCHER EXAMPLE: nal /max APPLICATION EXPLORER EXAMPLE: nalexpld /max	Displays the Application Launcher/ Explorer window maximized when first loaded, overriding the window state from the previous running of Application Launcher.

Switch	Description
SYNTAX: /min APPLICATION LAUNCHER EXAMPLE: nal /min APPLICATION EXPLORER EXAMPLE: nalexpld /min	Displays the Application Launcher/ Explorer window minimized when first loaded, overriding the window state from the previous running of Application Launcher/Explorer.
SYNTAX: /n APPLICATION LAUNCHER EXAMPLE: nal /n APPLICATION EXPLORER EXAMPLE: nalexpld /n	Displays the Application Launcher/ Explorer window in its normal state (neither maximized nor minimized) when first loaded, overriding the window state from the previous running of Application Launcher/Explorer.
SYNTAX: /nbc APPLICATION LAUNCHER EXAMPLE: nal /nbc APPLICATION EXPLORER EXAMPLE: nalexpld /nbc	Stands for No Boot Choice. When updating Application Launcher/ Explorer files, replaces the Yes/No buttons with an OK button when prompting the user to restart his or her workstation to update files in use.

Switch	Description
SYNTAX: /nodialup APPLICATION LAUNCHER EXAMPLE: nal /nodialup APPLICATION EXPLORER EXAMPLE: nalexpld /nodialup	Exits if a dial-up connection is present. Application Launcher/Explorer will not update any files nor run any other EXE if a dial-up connection is present.
SYNTAX: /noupdate APPLICATION LAUNCHER EXAMPLE: nal /noupdate APPLICATION EXPLORER EXAMPLE: nalexpld /noupdate	Does not update any Application Launcher/Explorer files on the workstation. Application Launcher/Explorer will be run from the current directory, usually SYS:PUBLIC.
SYNTAX: /nows APPLICATION LAUNCHER EXAMPLE: nal /nows APPLICATION EXPLORER EXAMPLE: nalexpld /nows	Instructs Application Launcher/Explorer to not register the workstation helper (ZENAPPWS.DLL) and to remove the registration if it exists. This switch should only be used for temporary troubleshooting purposes. If you want to permanently disable the workstation helper, you should disable the Enable Helper configuration option (see “Setting Configuration Options” on page 33).
SYNTAX: /ns EXAMPLE: nalexpld /ns	This switch is valid for Application Explorer only. The: switch provides the same functionality for Application Launcher. Skips the initial splash screen.

Switch	Description
SYNTAX: /p="*parameters*" APPLICATION LAUNCHER EXAMPLE: nal /a=.snapshot.apps /p="/a /q" APPLICATION EXPLORER EXAMPLE: nalexpld /a=.snapshot.apps /p="/a /q"	Instructs Application Launcher/ Explorer to pass the specified parameters to the application being launched. It is only effective when used with /a or /v.
SYNTAX: /s EXAMPLE: nal /s	This switch is valid for Application Launcher only. Instructs Application Launcher to act like the Windows shell. In the Application Launcher window, the Exit the Application Launcher option on the File menu changes to Shutdown Windows and gives the user the standard Windows shutdown options. **NOTE:** Application Launcher does not replace the Windows shell. If users minimize the Application Launcher window, they will have access to their normal desktop. If you want Application Launcher to replace the Windows shell, see "Using Application Launcher as the Windows Shell" on page 44.

Switch	Description
SYNTAX: /time=*seconds* APPLICATION LAUNCHER EXAMPLE: nal /time=10 APPLICATION EXPLORER EXAMPLE: nalexpld /time=10	Instructs Application Launcher/ Explorer to pause the specified number of seconds before continuing to load. The default wait time is 3 seconds. Only use this switch if you want Application Launcher/Explorer to wait extra time before continuing to load. For example, if several other programs are starting from the login script and they all display splash screens, you might not want all of them being displayed at the same time and overlapping each other.
SYNTAX: /u APPLICATION LAUNCHER EXAMPLE: nal /u APPLICATION EXPLORER EXAMPLE: nalexpld /u	Unloads Application Launcher/ Explorer from memory after all applications have terminated. **NOTE:** The quotation marks are required only if a space is used in the tree name or Application object name.

Switch	Description
SYNTAX: /v="*application_object*" or /v="*tree*:.*application_object*" APPLICATION LAUNCHER EXAMPLE: nal /v=".snapshot.apps" or nal /v="novell:.snapshot.apps" APPLICATION EXPLORER EXAMPLE: nalexpld /v=".snapshot.apps" or nalexpld /v="novell:.snapshot.apps"	Verifies and launches the specified application. *Application_object* is the full object name of the application to run. *Tree* is the name of the tree in which the object exists. The default tree is assumed if no tree name is specified.

5 Distributing Applications

The process of distributing an application consists of four main tasks:

1. Create the application's installation package and copy it to a network location.
2. Create an Application object in NDS® using the installation package.
3. Define the system requirements that a workstation must meet before Novell® Application Launcher™/Explorer will distribute the application to it.
4. Associate the Application object with the users or workstations you want to distribute the application to.

This process assumes that you have already set up Application Launcher/Explorer on users' workstations. Application Launcher/Explorer displays, installs, and configures the application on the workstation after you complete the tasks listed above. If you have not installed Application Launcher/Explorer, see Chapter 4, "Setting Up Application Launcher/Explorer," on page 27.

The following sections provide instructions for completing the tasks listed above:

- "Creating an Application's Installation Package" on page 55
- "Creating an Application Object" on page 57
- "Defining the Application's System Requirements" on page 61
- "Associating the Application Object with Users or Workstations" on page 63

HINT: Once you create an Application object, there are many different Application object property settings you can configure to better manage the application. The above steps provide information about the basic settings you must configure in order to distribute the application. For information about additional settings you can configure, see

Creating an Application's Installation Package

An installation package consists of the files and information needed to install the application on a workstation. The following sections explain the options available to you to help you create an application's installation package:

- "Manually Creating an Installation Package" on page 56
- "Creating a snAppShot Installation Package" on page 56
- "Creating a Microsoft Windows Installer Package" on page 57

Manually Creating an Installation Package

Some applications require few files to be copied to the workstation and few (or no) changes to the workstations configuration (registry, INI files, environment variables, and so forth). The installation package for these applications may very well consist of a one or two files only. In this case, you can copy the files to a network directory. After creating the NDS object for the application, you can configure the Application object to instruct Application Launcher/Explorer to simply run the application from the network directory or to first copy the files to the workstation and then run the application locally. You can also configure the Application object to modify any workstation settings required by the application.

To manually create an installation package:

1. Create a network directory for the application's files.
2. Copy the files to the network directory.

Creating a snAppShot Installation Package

Some applications require many files to be installed and many changes to be made to the workstation. This can require a complex installation package.

ZENworks™ for Desktops (ZfD) Application Management includes the snAppShot™ utility to automate the process of creating complex installation packages. You run snAppShot on a workstation that has never had the application installed to it. SnAppShot records the workstation's state (files and configuration) before you install the application and after you install the application. From this information, snAppShot creates an installation package consisting of an Application object template file (.AOT or .AXT file) and the application's files (.FIL files). This installation package is saved to a network directory you specify. When you create the NDS object for the application, the

.AOT file is used. When Application Launcher/Explorer distributes the application to a workstation, it uses the .FIL files as well as the configuration information (registry changes, INI file changes, and so forth) stored in the Application object.

To use snAppShot to create an installation package:

1. Set up a "clean" workstation. This needs to be a workstation to which the application has never been installed.
2. Run snAppShot (SNAPSHOT.EXE) from the SYS:\PUBLIC\SNAPSHOT directory.
3. Follow the on-screen prompts to create the installation package. For more detailed information about snAppShot, see Chapter 11, “SnAppShot,” on page 119.

Creating a Microsoft Windows Installer Package

Microsoft* Windows* 2000 (and patched versions of Windows 95/98 and Windows NT* 4.0) include an installation and configuration service referred to as Microsoft Windows Installer. Microsoft Windows Installer, similar to Application Launcher/Explorer, installs applications to a workstation from an installation package consisting of a .MSI file and various support files.

ZfD Application Management enables you to distribute applications that are installed through Windows Installer. Rather than using a snAppShot .AOT/.AXT file when creating the Application object, you simply use the application’s .MSI file. Distributing .MSI applications through Application Launcher/Explorer enables you to control who has access to the application while still gaining the benefits associated with the Windows Installer (such as on-demand installation, transforms, and source resiliency).

To use a Windows Installer package:

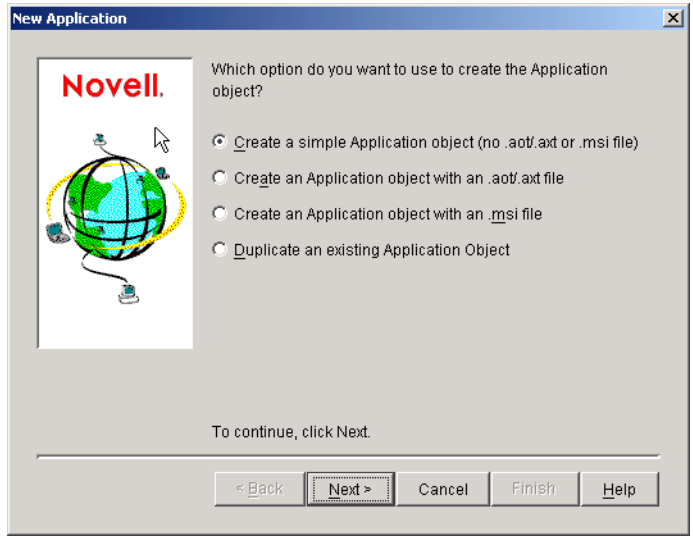
1. Create the installation package in a network location by running the application’s setup program using the administrative option. For more information, see you application’s installation documentation.

   or

   Copying the application’s preconfigured .MSI files and supporting files to a network location (if this method is supported by the application).

Creating an Application Object

1. In ConsoleOne™, right-click the container in which you want to create the Application object > click New > click Object to display the New Object dialog box.
2. Click App:Application > click OK to display the New Application dialog box.


New Application
Novell.
Which option do you want to use to create the Application object?
Create a simple Application object (no .aot/.axt or .msi file)
Create an Application object with an .aot/.axt file
Create an Application object with an .msi file
Duplicate an existing Application Object
To continue, click Next.
< Back
Next >
Cancel
Finish
Help

3. Select the option you want to use to create the Application object:

   Create a Simple Application Object (No .AOT/.AXT File or .MSI File): Lets you enter a path to the application's executable file. After the Application object is created, you can modify additional properties if necessary. Skip to "Creating a Simple Application Object" on page 58.

   Create an Application Object With an .AOT/.AXT File: Lets you specify an .AOT or .AXT file you've created with snAppShot. The .AOT or .AXT file is used to populate the Application object's property fields. Skip to "Creating an Application Object With an .AOT/AXT File" on page 59.

   Create an Application Object With an .MSI File: Lets you specify an .MSI file. The .MSI file is used to populate the Application object's property fields. Skip to "Creating an Application Object With an .MSI File" on page 60.

Duplicate an Existing Application Object: Lets you make a copy of an existing Application object. Skip to "Duplicating an Application Object" on page 61.

Creating a Simple Application Object

1. In the New Application dialog box, select Create a Simple Application Object (No .AOT/.AXT File or .MSI File) > click Next.
2. In the Object Name box, type a name for the Application object.
3. In the Path to Executable box, enter the path to the location from which the application's executable file will be run. Include the executable file in the path.

   IMPORTANT: If you plan for users to run the application from the network or the file is already located on the users' workstations, the path should be to the actual executable file on the network or the workstation. If you plan to have Application Launcher/Explorer copy the file to a target directory on the workstation, the path must be to the executable file in the target directory.
4. Click Finish to create the Application object.
5. If Application Launcher/Explorer will need to copy files to the workstation, right-click the Application object > click Properties > complete the remaining steps.

   or

   If the application will be run from the network or is already located on the workstation, skip the remain steps and continue with "Defining the Application's System Requirements" on page 61.
6. Click the Distribution Options tab > click Application Files.

   You need to add the files to the Application Files list in order for Application Launcher/Explorer to copy them to the workstation.
7. Click Add > click File > fill in the following fields.

   Source File: Select the file you want copied to the workstation (for example, j:\public\notepad\notepad.exe).

   Target File: Enter the full path, including the file name, of where you want the file copied (for example, c:\notepad\notepad.exe).
8. Click OK to add the file to the list.
9. Repeat Step 7 and Step 8 for each file to be copied.

10 When finished adding files, click OK to save the Application object information.

11 Continue with "Defining the Application's System Requirements" on page 61 to define the application's system requirements.

Creating an Application Object With an .AOT/AXT File

1 In the New Application dialog box, select Create an Application Object With an .AOT/AXT File > click Next.

2 Enter the path to the .AOT or .AXT file > click Next.

HINT: The file should be in the network location you saved it to when creating it with snAppShot.

3 If necessary, modify the following fields to customize the Application object.

Object Name: This field defaults to the Application object name that was specified when running snAppShot to create the installation package. You can change the name if you want. If another Application object in the same container has the same name, you need to select a different name. No two objects in the same container can have the same name.

SOURCE_PATH: This field defaults to the location where the application's installation files (.FIL) files where stored when running snAppShot to create the installation package. You should verify that the path is correct.

TARGET_PATH: This field defaults to the location where the application was installed when running snAppShot to create the installation package. You should verify that this is the workstation directory where you want the application installed.

4 Click Next.

5 Review the Application object settings > click Finish to create the Application object.

6 Continue with "Defining the Application's System Requirements" on page 61 to define the application's system requirements.

Creating an Application Object With an .MSI File

1 In the New Application dialog box, select Create an Application Object With an .MSI File > click Next.

2 Enter the path to the .MSI file > click Next.

HINT: The file should be in the network location where you installed or copied the application.

3 If necessary, modify the following fields to customize the Application object.

Object Name: This field defaults to the Application object name that was specified when running snAppShot to create the installation package. You can change the name if you want. If another Application object in the same container has the same name, you need to select a different name. No two objects in the same container can have the same name.

SOURCE_PATH: This field defaults to the administrative installation location defined in the .MSI file. You should verify that the path is correct.

4 Click Next.

5 Review the Application object settings > click Finish to create the Application object.

6 Continue with "Defining the Application's System Requirements" on page 61 to define the application's system requirements.

Duplicating an Application Object

1 In the New Application dialog box, select Duplicate an Existing Application Object > click Next.

2 Browse for and select the Application object you are duplicating > click Next.

3 Fill in the following fields to customize the new Application object.

Object Name: Specify a name for the new Application object.

SOURCE_PATH: Specify the network location where the application's installation files are located.

TARGET_PATH: Specify the workstation location where you want to install the application.

4 Click Next.

5 Review the Application object settings > click Finish to create the Application object.

6 Continue with "Defining the Application's System Requirements" on page 61 to define the application's system requirements.

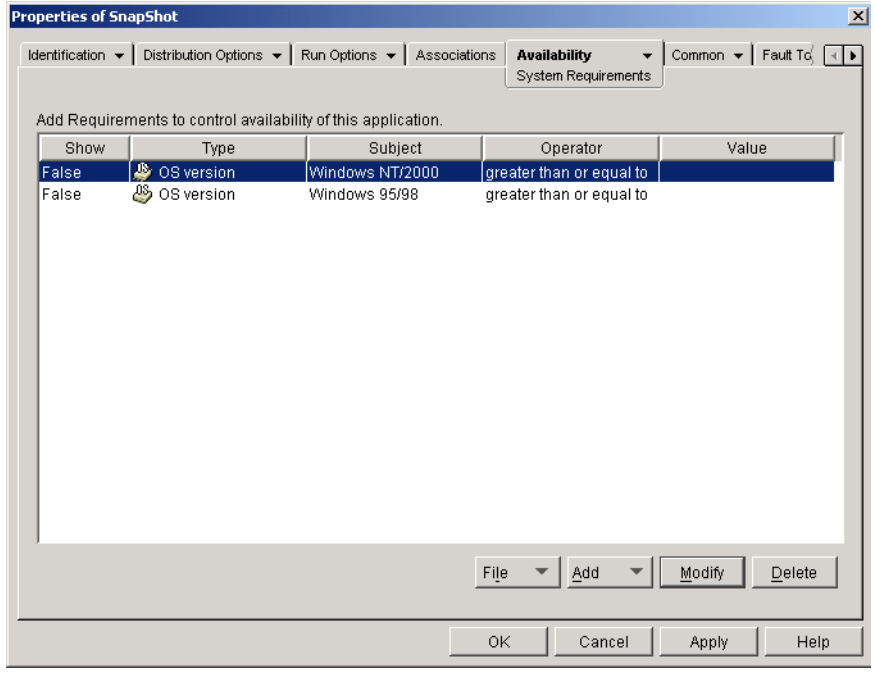
Defining the Application's System Requirements

To ensure that Application Launcher/Explorer does not distribute the application on a workstation that cannot support the application, you need to define the application's system requirements. An application will not even be displayed on users' workstations until you do so.

For example, the application may run on Windows 95/98 but not on Windows NT/2000. In this case, you would want to set one of the system requirements to be Windows 95/98. Or, the application may require 100 MB of free disk space, in which case you could establish that as a system requirement.

1 In ConsoleOne, right-click the Application object > click Properties.

2 Click the Availability tab > click System Requirements to display the System Requirements page.



3 Verify that the list contains an OS Version requirement (Windows NT/2000 or Windows 95/98) for the workstation you want to distribute the application to. If the correct OS version is not listed, click Add > Operating System > enter the requirements for the operating system > click OK to add it to the list.

IMPORTANT: An OS Version requirement must be defined before an application will be distributed.

4 Define any additional requirements you want.

For information about each requirement type, see "System Requirements Page" on page 169.

5 Click OK to save the changes.

Associating the Application Object with Users or Workstations

For Application Launcher/Explorer to display the application on a workstation, you must associate the Application object with the user logged in at the workstation or with the workstation itself.

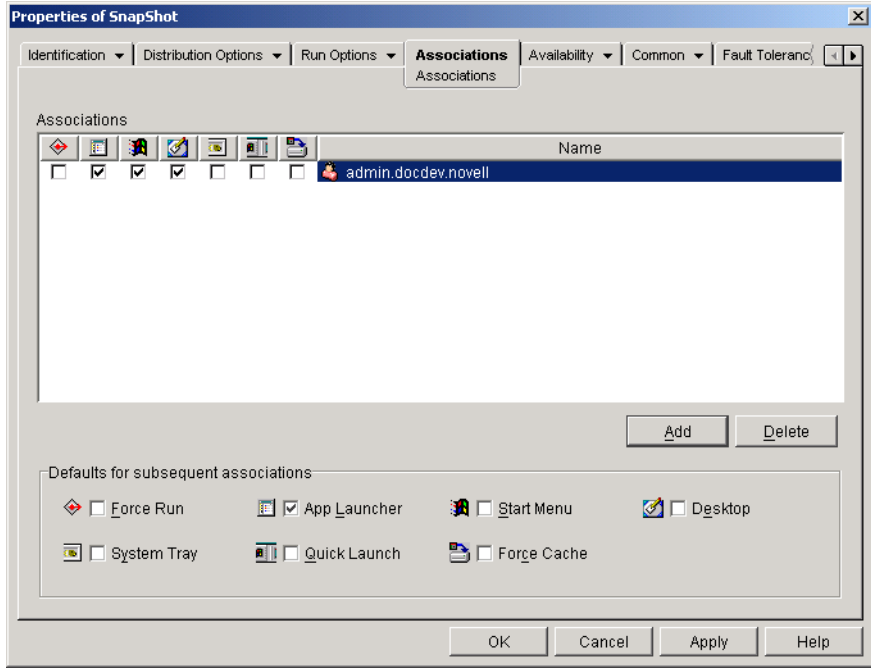
- "Associating the Application Object with Users" on page 63
- "Associating the Application Object with Workstations" on page 65

Associating the Application Object with Users

You associate the application with a user by adding the User object, a Group object to which the user belongs, or a container object (Organizational Unit, Organization, or Country) in which the user resides to the Application object's association list.

1 In ConsoleOne, right-click the Application object > click Properties.

2 Click the Associations tab to display the Associations page.



3 Click Add > browse and select a User object, Group object, or container object > click OK to add the user, group, or container to the Associations list.

4 Select the options you want applied to the Application object. By default, the App Launcher option is already checked, which causes the Application object to be displayed in the Application Launcher/Explorer window.

Force Run: Runs the Application object as soon as Application Launcher/Explorer or Workstation Helper starts and the Application object is available.

You can use the Force Run option in conjunction with several other settings to achieve unique behaviors. For example, if you use the Force Run option with the Run Application Once option (Run Options tab > Applications page), as soon as the Application object is distributed it will be run one time and then removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a pre-determined time. If so, select Force Run, select the Run Application Once option on the Application page (Run Options tab), and define a schedule using the Schedule page (Availability tab).

If you want to force run several applications in a specific order, mark each of them as Force Run. Then give each Application object a numeric order by using the Order Icon Display option on the Icon page (Identification tab).

IMPORTANT: When associating an Application object with a workstation, Force Run will not work if the Application object uses a Microsoft Windows Installer (.MSI) package. This is true only when the Application object is associated with a workstation. Force Run works with .MSI Application objects associated with users.

App Launcher: Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu: If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be added to the top of the Start menu, unless you assign the Application object to a folder and use the folder structure on the Start menu. See the Application object's Folders page (Identification tab).

Desktop: If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows desktop.

System Tray: If the workstation is running Application Explorer, this option displays the Application object's icon in the Windows system tray.

Quick Launch: Displays the Application object's icon on the Windows Quick Launch toolbar.

Force Cache: Forces the application source files and other files required for installation to be copied to the workstation's cache directory. The user can then install or repair the application while disconnected from NDS. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see Chapter 7, "Running in Disconnected Mode," on page 75.

IMPORTANT: This option is required only if you want to ensure that the user can install or repair the application while disconnected from NDS. Without this option selected, the user will still be able to launch the application in disconnected mode, provided the application has already been distributed (installed) to the workstation.

5 Click OK to save the Application object information.

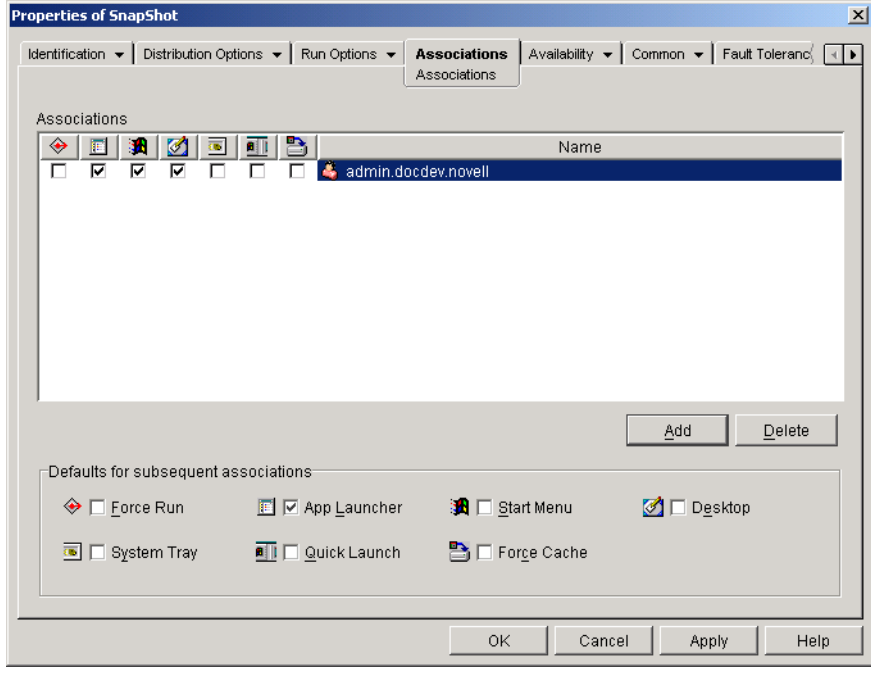
Associating the Application Object with Workstations

You associate the application with a workstation by adding the Workstation object, a Workstation Group object to which the workstation belongs, or a

container object (Organizational Unit, Organization, or Country) in which the workstation resides to the Application object's associations list.

IMPORTANT: Each workstation that will be associated with applications must first be imported into NDS as a Workstation object. For details about importing workstations, see Automatic Workstation Import in *Deployment*.

1. In ConsoleOne, right-click the Application object > click Properties.
2. Click the Associations tab to display the Associations page.



3. Click Add > browse and select a Workstation object, Workstation Group object, or container object > click OK to add the workstation, group, or container to the Associations list.
4. Select the options you want applied to the Application object. By default, the App Launcher option is already checked, which causes the Application object to be displayed in the Application Launcher/Explorer window.

   Force Run: Runs the Application object as soon as Application Launcher/Explorer or Workstation Helper starts and the Application object is available.

You can use the Force Run option in conjunction with several other settings to achieve unique behaviors. For example, if you use the Force Run option with the Run Application Once option (Run Options tab > Applications page), as soon as the Application object is distributed it will be run one time and then removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a pre-determined time. If so, select Force Run, select the Run Application Once option on the Application page (Run Options tab), and define a schedule using the Schedule page (Availability tab).

If you want to force run several applications in a specific order, mark each of them as Force Run. Then give each Application object a numeric order by using the Order Icon Display option on the Icon page (Identification tab).

IMPORTANT: When associating an Application object with a workstation, Force Run will not work if the Application object uses a Microsoft Windows Installer (.MSI) package. This is true only when the Application object is associated with a workstation. Force Run works with .MSI Application objects associated with users.

App Launcher: Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu: If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be added to the top of the Start menu, unless you assign the Application object to a folder and use the folder structure on the Start menu. See the Application object's Folders page (Identification tab).

Desktop: If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows desktop.

System Tray: If the workstation is running Application Explorer, this option displays the Application object's icon in the Windows system tray.

Quick Launch: Displays the Application object's icon on the Windows Quick Launch toolbar.

Force Cache: Forces the application source files and other files required for installation to be copied to the workstation's cache directory. The user can then install or repair the application while disconnected from NDS. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see Chapter 7, "Running in Disconnected Mode," on page 75.

IMPORTANT: This option is required only if you want to ensure that the user can install or repair the application while disconnected from NDS. Without this option selected, the user will still be able to launch the application in disconnected mode, provided the application has already been distributed (installed) to the workstation.

5 Click the NDS Rights tab > click Trustees of This Object.

Associating the Application object with the workstation enables Application Launcher/Explorer to display the object on the workstation. Any users who will run the application on the workstation must be trustees of the Application object. Trustee rights are not granted automatically; you must manually give default trustee rights to each user.

6 Click Add Trustee > select the user you want to add as a trustee > click OK to add the user to the trustee list. Repeat this step for all users you need to add.

7 Click OK to save the Application object information.

Other Things You Can Do

To better manage the application you've distributed, you can configure the Application object to:

- Automatically cache the application's installation files to the local workstation. This enables users to install or fix problems with the application when they are disconnected from NDS and your network. See Chapter 7, "Running in Disconnected Mode," on page 75.
- Add the Application object to a folder in the Application Launcher/Explorer window or on the Windows Start menu. See Chapter 8, "Organizing Applications into Folders," on page 89.
- Report on the success or failure of events such as distributing, caching, and uninstalling the application on workstations. See Chapter 9, "Reporting on Application Management Events," on page 97.
- Track the number of licenses being used for the application. See Chapter 10, "Metering Software Licenses," on page 115.
- Determine the schedule of when the application will be available to users. See "Schedule Page" on page 184.
- Set up backup copies of the application source files to enable fault tolerance. See "Fault Tolerance Page" on page 210.
- Define multiple Application objects for the same application and enable load balancing. See "Load Balancing Page" on page 212.

- Set up site lists to enable users who travel from site to site to continue to have access to their applications. See “Site List Page” on page 214.
- Define many additional Application object properties to control the behavior of the application. For information about all Application object properties, see Chapter 12, “Application Object Settings,” on page 123.

6 Uninstalling Applications

Any application (including MSI-based applications) distributed through ZENworks™ for Desktops (ZfD) Application Management can be uninstalled. You can choose whether or not to uninstall all files, shortcuts, INI entries, and registry entries that were installed or modified with the application.

Each user's workstation contains a local cache that contains information about applications installed on the workstation. When you uninstall an application, this cache is used to ensure that the appropriate files and settings are removed from the workstation. If the application's source files have been copied to the workstation's local cache, the source files are also removed. For more information about the local cache, see Chapter 7, "Running in Disconnected Mode," on page 75.

By default, the ability to uninstall an application is disabled. You must first enable an application to be uninstalled. After that, you can uninstall an application by disassociating it with users or workstations, you can configure an application to be uninstalled after a certain period of unuse, or your users can use Novell® Application Launcher™/Explorer to uninstall an application (provided you've enabled this functionality). See the following sections for information:

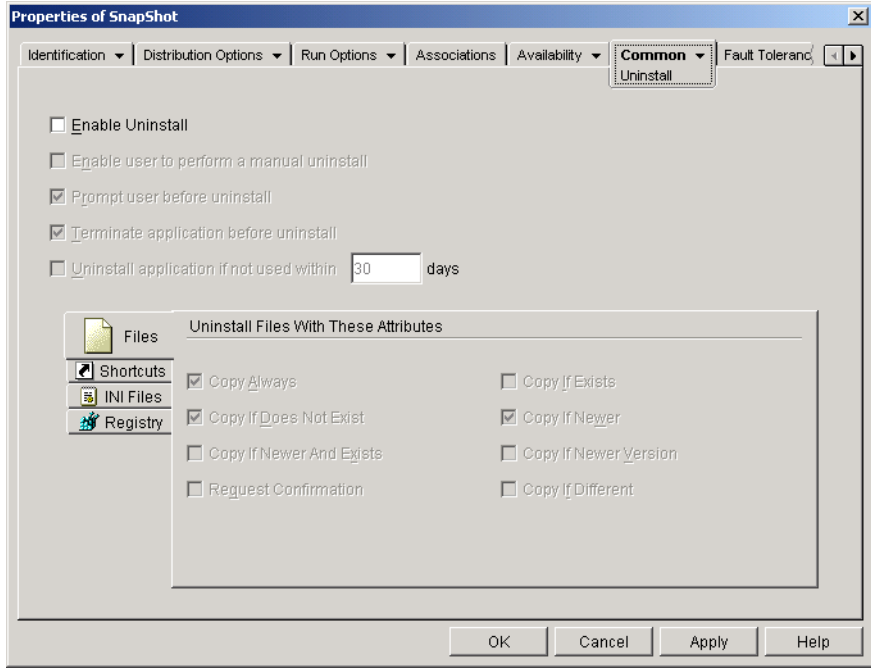
- "Enabling an Application to be Uninstalled" on page 69
- "Uninstalling Applications by Disassociating Users or Workstations" on page 72
- "Uninstalling Unused Applications" on page 73
- "Using Application Launcher/Explorer to Uninstall an Application" on page 74

Enabling an Application to be Uninstalled

Uninstalling an application requires careful configuration of the Application object to ensure that no critical files are accidentally removed from the workstation. For this reason, the ability to uninstall an application is disabled by default.

To enable an application to be uninstalled:

1. In ConsoleOne, right-click the Application object > click Properties.
2. Click the Common tab > click Uninstall to display the Uninstall page.



3. Click the Enable Uninstall box to turn on the option.
4. If you want users to be able to uninstall the application themselves, click the Enable User to Perform a Manual Uninstall to turn on the option.

   For information about how users can uninstall the application, see "Using Application Launcher/Explorer to Uninstall an Application" on page 74.
5. Set the other Uninstall options as appropriate:

   Prompt User Before Uninstall: Select this option to prompt users before removing the application from their workstations. If users answer "No" to the prompt, the application will not be removed.

   Terminate Application Before Uninstall: Select this option to ensure that Application Launcher/Explorer terminates the application (if the application is running) before it begins uninstalling the application files.

   Uninstall Application If Not Used Within *XX* Days: Select this option to automatically remove the application if the user has not run it within

the specified number of days (the default is 30). For more information, see “Uninstalling Unused Applications” on page 73.

6 Select the attributes you want used when determining whether or not to uninstall files, shortcuts, INI settings, and registry settings.

Files: Click Files > check the attributes you want to use to determine whether or not an application file is removed. The attributes correspond to the attributes listed on the Application Files page (Distribution Options tab). By default, the options are configured (Copy Always, Copy if Does Not Exist, Copy if Newer) to ensure that a file is removed only if it was installed when the application was installed and not at an earlier time by another application that also needs the file.

Shortcuts: Click Shortcuts > check the attributes you want to use to determine whether or not the application's shortcuts are removed. The attributes correspond to the attributes listed on the Shortcuts/Icons page (Distribution Options tab).

INI Files: Click INI Files > check the attributes you want to use to determine whether or not an INI setting is removed. The attributes correspond to the attributes listed on the INI Settings page (Distribution Options tab). By default, the options are configured (Create Always, Create if Exists, Create or Add to Existing Section) to ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that also needs the setting.

Registry: Click Registry > check the attributes you want to use to determine whether or not a registry setting is removed. The attributes correspond to the attributes listed on the Registry Settings page (Distribution Options tab). By default, the options are configured (Create Always, Create if Exists) to ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that still needs the setting.

The Uninstall Keys/Values From These Registry Hives option allows you to select two specific registry hives, HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER, to include or exclude when removing the registry settings that were added by Application Launcher/Explorer during distribution of the application to the workstation. The default settings cause Application Launcher/Explorer to remove settings from the HKEY_CURRENT_USER hive but not from the HKEY_LOCAL_MACHINE hive.

The purpose of this option is to help you ensure that no Windows* system settings are removed during the uninstall. This is particularly important if

you used snAppShot™ to capture the application's installation. When snAppShot captures an application's installation, it includes all settings that are changed during the installation. If the application's installation program causes the workstation to reboot during the installation process, these changes can include not only settings that apply to the application but also settings that apply to the Windows system. Removing Windows system settings may cause the workstation to encounter problems when starting up. It is strongly recommended that you remove these settings only if you are positive that they will not affect the workstation's ability to function properly.

NOTE: If you want to use this option, you should examine the registry settings that will be removed from the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER hives and delete any settings that apply to anything other than the application. The registry settings are displayed on the Registry Settings page (Distribution Options tab).

7 Click OK to save the changes.

Uninstalling Applications by Disassociating Users or Workstations

You can force an application to be uninstalled by disassociating it from the user or workstation, provided you have enabled the application to be uninstalled (see "Enabling an Application to be Uninstalled" on page 69). If you disassociate an application that has not been configured as uninstallable, the application's files, shortcuts, INI settings, and registry settings will not be removed from the workstation.

1 In ConsoleOne, right-click the Application object > click Properties.

2 Click the Associations tab.

3 Select the user or workstation you want to disassociate from the application > click Delete.

4 Click OK.

The next time Application Launcher/Explorer refreshes, it will uninstall the application from the user's workstation. In addition, the Application object will no longer be displayed on the user's workstation.

HINT: You can use the Unassociated Days to Uninstall option to specify a longer grace period for the user. To do so, right-click the user's object in ConsoleOne > click Properties > click the Application Launcher tab > click Edit > modify the

Unassocaited Days to Uninstall option. For additional information, see "Configuring Application Launcher/Explorer" on page 33.

Uninstalling Unused Applications

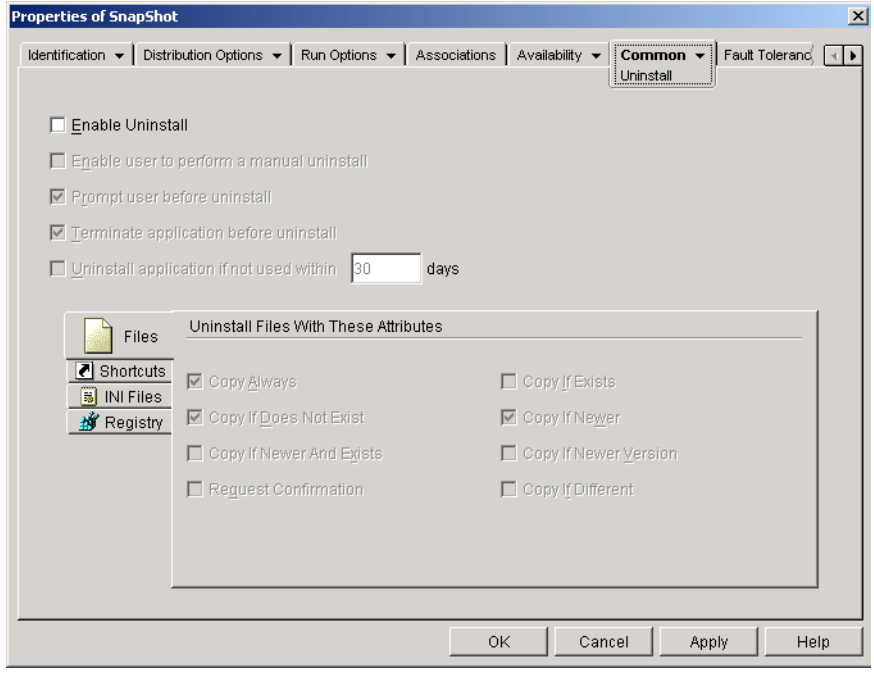
Users may install some applications, run them a few times, and then not use them again for a long period of time. These applications can unnecessarily take up valuable workstation disk space.

To free up disk space, you can have Application Launcher/Explorer uninstall an application that has not been run for a period of time. You can determine the time period; the default is 30 days. For example, you may want one application removed after 15 days of unuse, while you may not want to remove another application until it has not been run for 90 days.

Unless the user or workstation is disassociated from the Application object, Application Launcher/Explorer will continue to display the Application object icon on the workstation after the application is uninstalled. If the user double-clicks the icon, the application will be installed again.

To configure an unused application to be removed after a period of time:

1. In ConsoleOne, right-click the Application object > click Properties.
2. Click the Common tab > click Uninstall to display the Uninstall page.



3 If the Enable Uninstall option is not turned on, click the box to turn it on.

4 Click the Uninstall Application If Not Used Within *XX* Days box to turn on the option > select the number of days you want.

5 If you have not selected the attributes you want to use to determine whether or not the application's files, shortcuts, INI settings, and registry settings are removed, do so now. For information about each option, see "Enabling an Application to be Uninstalled" on page 69.

6 Click OK to save the changes.

Using Application Launcher/Explorer to Uninstall an Application

By default, users are not allowed to uninstall applications that you've distributed to them. However, if you've enabled users to uninstall applications (see "Enabling an Application to be Uninstalled" on page 69), they can do so through Application Launcher/Explorer.

1 On the user's workstation, right-click the Application object > click Uninstall.

When Application Launcher/Explorer uninstalls an application, it removes all application files and settings from the workstation. However, if the Application object is still associated with the user, the Application object's icon will remain on the workstation. This enables the user to reinstall the application at a later time.

7 Running in Disconnected Mode

Novell® Application Launcher™/Explorer enables users to install, run, verify (self-heal), and uninstall applications while disconnected from NDS® and the network. This enables users to run the same applications when connected to or disconnected from NDS.

The following sections provide information and instructions to help you understand the Application Launcher/Explorer disconnected mode and to support disconnected users:

- "Disconnectable Applications" on page 75
- "Workstation Cache" on page 76
- "Configuring Application Launcher/Explorer to Start Automatically" on page 79
- "Configuring Applications as Disconnectable" on page 81
- "Forcing an Application to be Cached" on page 82
- "Changing the Default Cache Location" on page 83
- "Distributing Applications to Disconnected Workstations" on page 84
- "Enabling Users to Manage Applications" on page 85

Disconnectable Applications

When you create an Application object, the application is automatically configured as being disconnectable. Once a disconnectable application is distributed (installed) to a workstation, Application Launcher/Explorer will continue to display the Application object icon even after the workstation has disconnected from NDS and will attempt to launch the application when the user double-clicks the Application object icon.

Only applications that can run locally on a disconnected workstation should be configured as disconnectable. Applications that rely on network resources should not be configured as disconnectable. This includes:

- Applications that access a network database
- Client/server applications
- Applications that depend on network drive mappings or print captures

- Applications that use NDS macros (other than %CN%) in any of the Application object property fields
- Applications that require a persistent connection to the network

Workstation Cache

To support disconnected mode, Application Launcher/Explorer creates a cache directory on the workstation's Windows* drive (the drive containing the SYSTEM32 directory). This directory, called NALCACHE, is created as a hidden directory at the root of the drive. If necessary, the location can be changed (see "Configuring Application Launcher/Explorer" on page 33).

The NALCACHE directory contains subdirectories for each application associated with the user. Each application's folder contains the following two types of cache:

- "Launch Cache" on page 76
- "Install Cache" on page 78

Launch Cache

Application Launcher/Explorer automatically creates an application's launch cache when the application is distributed to a user or workstation. The launch cache contains all the NDS information Application Launcher/Explorer needs to launch and run an application while disconnected. This cache also contains the information needed to uninstall the application.

The launch cache files are stored in the NALCACHE*NDS_TREE**APP_OBJECT* directory, where *APP_OBJECT* is the NDS context of the Application object and *NDS_TREE* is the tree where the Application object resides. By default, all files except those containing application files, INI settings, shortcut icons, and registry settings are refreshed each time Application Launcher/Explorer performs a timed or manual refresh. These files remain the same until the Version Number (Application object > Distribution Options tab > Options page) is changed so that the application can be successfully uninstalled.

The following table explains the files that may reside in the launch cache. An asterisk next to a file name indicates that Application Launcher/Explorer does not update that file when it refreshes.

File	Description
DSATTR.BIN	Contains all of the Application object's NDS attributes.
FOLDERS.BIN	Contains the list of folders where the Application object will be displayed in the Application Launcher/Explorer window and the Start menu.
COMPLETE.BIN	Contains the Application object's complete NDS name. If this file does not exist, Application Launcher/ Explorer assumes that the cache is bad and recreates the cache.
STRM1.BIN	Contains the Application object's icon.
STRM2.BIN	Contains all text file modifications.
STRM3.BIN*	Contains all files to be copied to the workstation.
STRM4.BIN*	Contains registry modifications that will be made during initial distribution.
STRM5.BIN*	Contains INI modifications that will be made during initial distribution.
STRM6.BIN*	Contains shortcut modifications that will be made during initial distribution.
STRM7.BIN	Contains macro information.
STRM8.BIN	Contains schedule information.
STRM9.BIN	Contains system requirements information.
STRM10.BIN	Contains Administrator notes.
STRM11.BIN	Contains shutdown script.
STRM12.BIN	Contains startup script.

File	Description
STRM13.BIN*	Contains registry modifications to be made each time the application is launched.
STRM14.BIN*	Contains INI modifications to be made each time the application is launched.
STRM15.BIN*	Contains files to be copied to the workstation each time the application is launched.
STRM16.BIN	Contains text file modifications to be made each time the application is launched.
STRM17.BIN*	Contains icons to be modified each time the application is launched.
STRM18.BIN	Contains environment variable settings to be made.
STRM19.BIN	Contains the pre-distribution script.
STRM20.BIN	Contains the post-distribution script.
STRM21.BIN	Contains...

Install Cache

Application Launcher/Explorer creates an application's install cache only if you select the Force Cache option when associating the Application object with users or workstations, or if individual users use Application Launcher/Explorer's Application Management feature to cache the application. The install cache contains the application source files needed to install or verify an application while disconnected. While the files are compressed, they can still require a large amount of disk space on the workstation local drive, so an application should only be cached if users will need to install the application or verify (fix) the already installed application while disconnected.

IMPORTANT: An application can still be run in disconnected mode without being cached to the install cache. To run in disconnected mode, an application needs only to have been distributed (installed) on the workstation. Caching an application to the install cache only enables the application to be installed while in disconnected mode, or to be verified (fixed) if a problem occurs.

Configuring Application Launcher/Explorer to Start Automatically

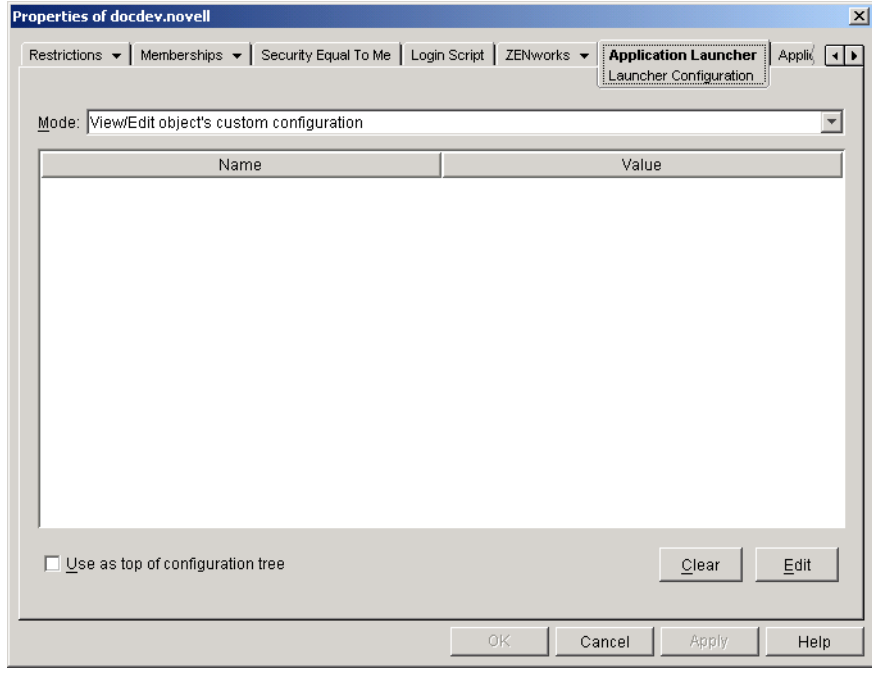
Typically, when connected to the network, Application Launcher/Explorer is started through a command in the user's login script. To automatically start Application Launcher/Explorer in disconnected mode, you can add Application Launcher or Application Explorer to the Windows Startup folder by using the Launcher Configuration settings in NDS.

Depending on which program is running at the time, either an Application Launcher (NALWIN32.EXE) shortcut or an Application Explorer (NALDESK.EXE) shortcut is added to the Startup folder.

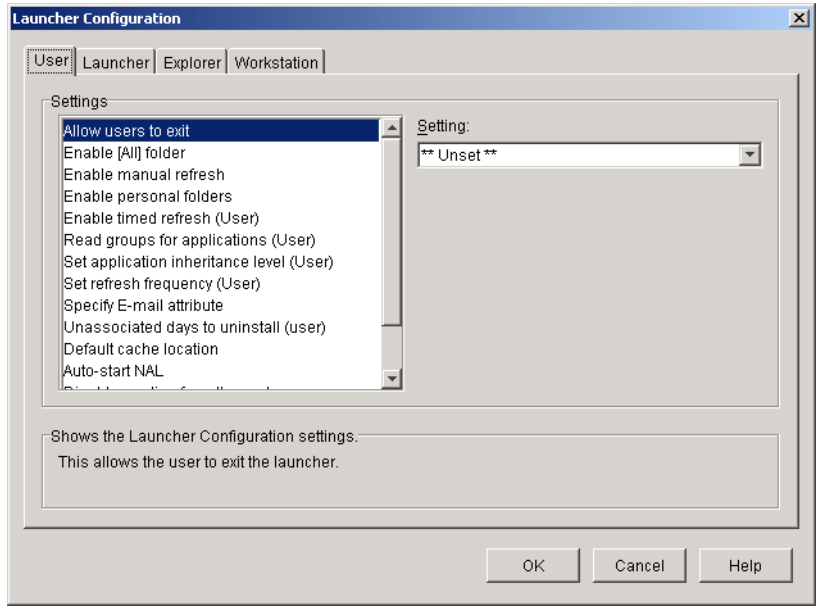
If Application Launcher or Application Explorer attempts to start twice, once through the user's login script and once through the Startup folder, only one instance of the program will be launched.

To add Application Launcher or Application Explorer to the Startup folder:

1. In ConsoleOne™, right-click the User, Group, or container object for the users whose Application Launcher/Explorer you want to configure > click Properties.
2. Click the Application Launcher tab to display the Launcher Configuration page.



3 Click Edit to display the Launcher Configuration dialog box.



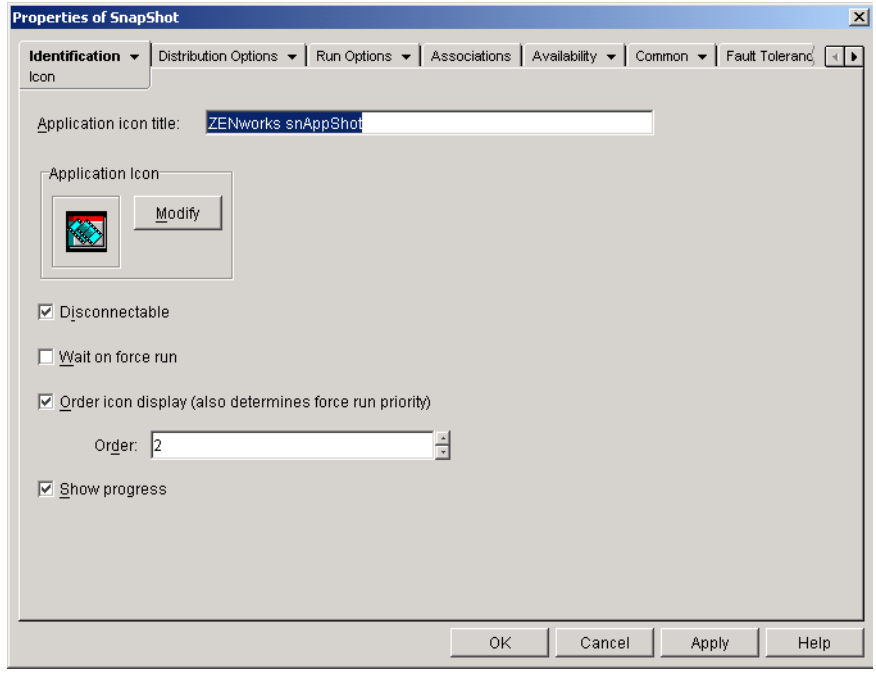
4 In the Settings list (User tab), select Auto-Start NAL > set the option to Yes.

5 Click OK > click OK to save the changes.

Configuring Applications as Disconnectable

By default, when you create an Application object the application is marked as being disconnectable. To verify this setting or mark the application as not being disconnectable:

1 In ConsoleOne, right-click the Application object > click Properties.



2 Select the Disconnectable option to make the application disconnectable.

or

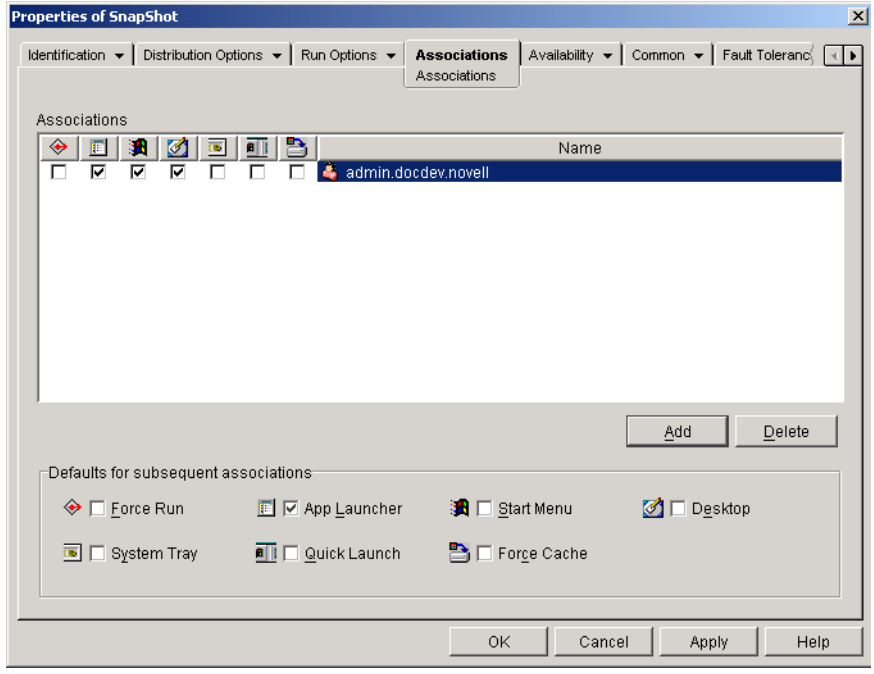
Deselect the option to make the application not disconnectable.

3 Click OK to save the changes.

Forcing an Application to be Cached

If you want to ensure that a user has an application cached to his or her local drive so that he or she can install or verify the application while disconnected, you can force cache the application. For information about enabling users to cache applications themselves, see “Enabling Users to Manage Applications” on page 85.

1. In ConsoleOne, right-click the Application object > click Properties.
2. Click the Associations tab to display the Associations page.



3. In the Associations list, select the Force Cache box for the user whose workstation you want to cache the application to.
4. Click OK to save the changes.

HINT: The same task can be accomplished through the Applications tab on a User, Workstation, Group, Workstation Group, or container object.

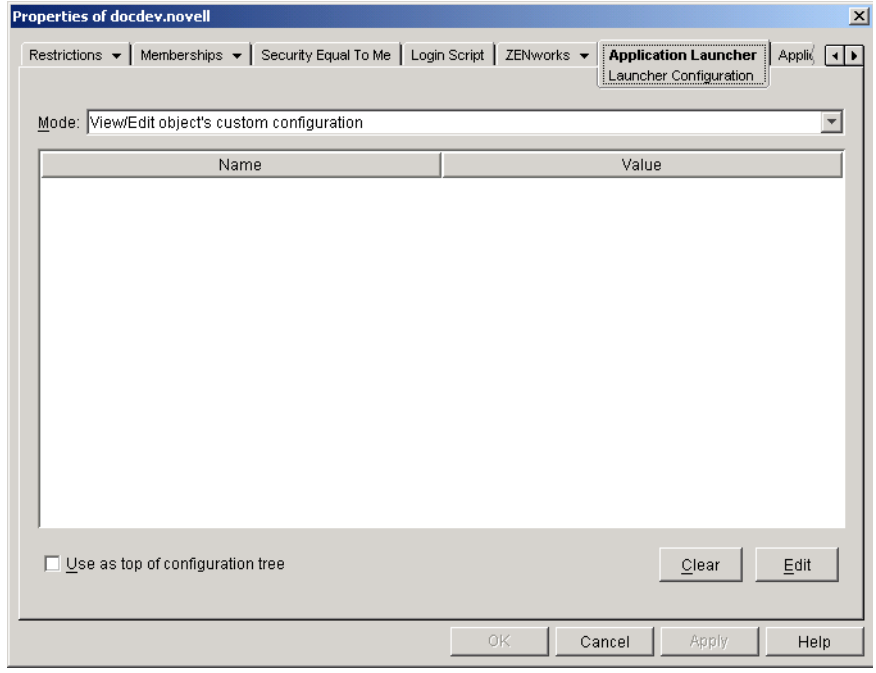
Changing the Default Cache Location

As mentioned in “Workstation Cache” on page 76, the default cache (NALCACHE) is created as a hidden directory on the root of the workstation’s Windows drive.

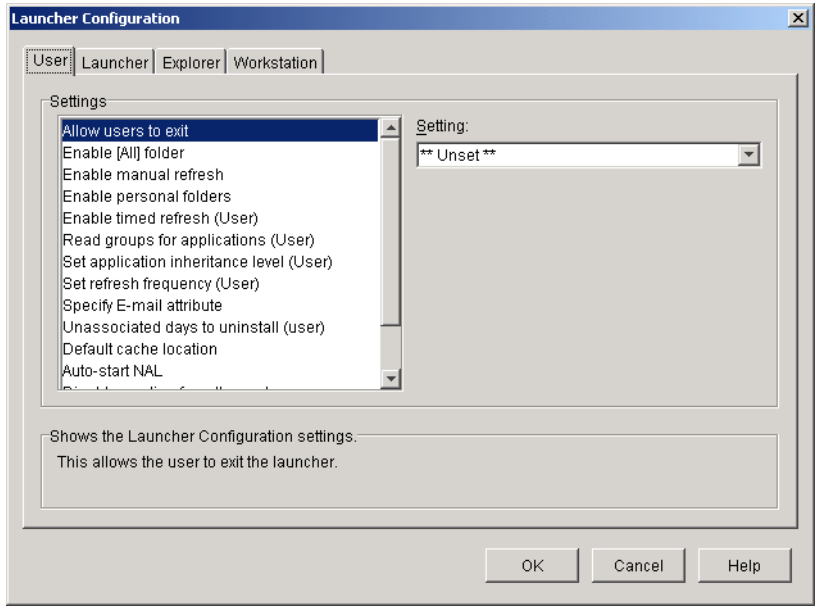
You can change the location by specify a different drive letter. The next time Application Launcher/Explorer refreshes (either a timed refresh or a manual refresh), the NALCACHE directory will be recreated at the root of the location that the workstation’s drive letter is mapped to. The old NALCACHE directory is not removed automatically.

To change the location of the cache directory:

1. In ConsoleOne, right-click the User, Group, or container object for the users whose cache location you want to change > click Properties.
2. Click the Application Launcher tab to display the Launcher Configuration page.



3. Click Edit to display the Launcher Configuration dialog box.



4 In the Settings list (User tab), select Default Cache Location > select the drive letter you want to use.

5 Click OK > click OK to save the changes.

Distributing Applications to Disconnected Workstations

As long as a disconnected workstation has Application Launcher/Explorer installed and running, you can distribute applications to the workstation through the use of removable media such as a CD, JAZ* disk, or ZIP* disk. The removable media functions as a second workstation cache, containing the Application object settings and application source files required to install and run the application.

For example, you have a mobile user who seldom connects to the network but needs an application that you've distributed to other users. You burn a CD with the application and then send it to the user. The user inserts the CD into his or her workstation's drive, Application Launcher/Explorer reads the CD and displays the Application object's icon in the places you've configured (Application Launcher/Explorer window, Start menu, desktop, and so forth). The user then launches the application, which is distributed to the workstation according to the Application object's configuration.

NOTE: The removable media includes an AUTORUN.INF file that causes Application Launcher/Explorer to refresh and display the Application object. If the user's workstation is not configured to autorun CDs, or you use another media type, the user may have to manually refresh Application Launcher/Explorer. This is done by right-clicking the Application Launcher/Explorer icon and clicking Refresh.

To create a cache on a removable media:

1. In ConsoleOne, click the Tools menu > click Application Launcher Tools > click Create Virtual CD to start the Create CD Wizard.
2. Click Add > browse for and select the Application object for the application you want to distribute. Repeat this step to add additional applications.
3. For each application you added to the list, select the locations where you want the Application object's icon to appear and select whether or not you want to force run and force cache the application. For information about each of these options, see "Associations Page" on page 166.
4. After you've finished adding applications, click Next.
5. In the Select Directory for Virtual CD, select the location where you want to create the cache for the applications.

   IMPORTANT: Make sure the location you select has sufficient disk space for the application cache. If enough disk space does not exist, you'll receive a message stating that an error has occurred.
6. If you want to restrict the amount of time the user can continue to use the application (or applications) after the media has been removed from the workstation, select the When CD Is Removed Disassociation Will Occur After option > select the number of days after which the disassociation will occur.

   The time is measured from when the CD (or removable media) is removed from the user's workstation. The default time is 30 days. The valid range is 0 to 65535 days.
7. Click Next > review the settings > click Finish to create the cache.
8. If necessary, copy the cache to the removable media (for example, burn the CD).

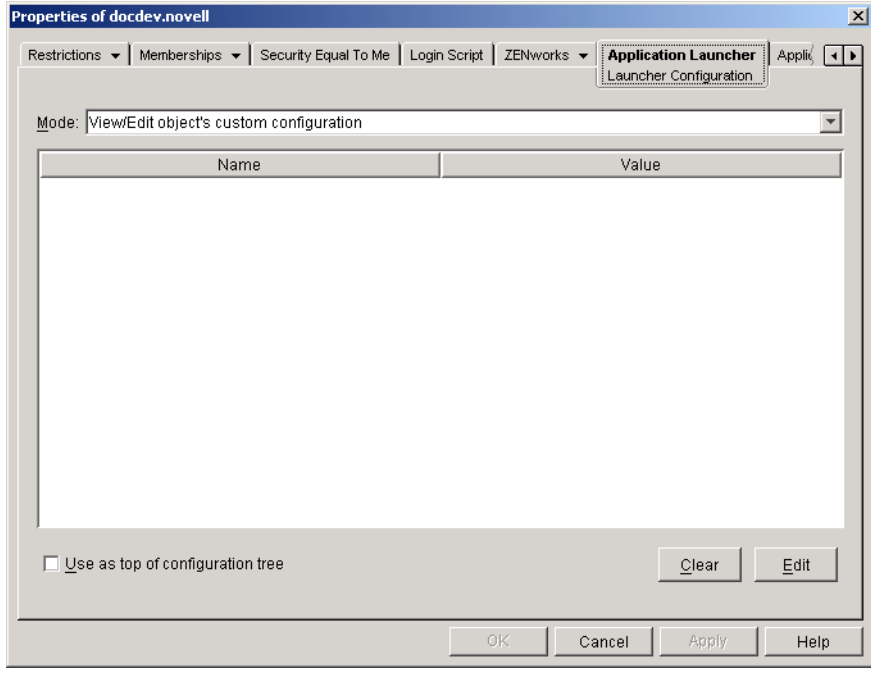
Enabling Users to Manage Applications

Application Launcher/Explorer includes a Manage Applications feature that enables users to cache applications, remove applications from the cache,

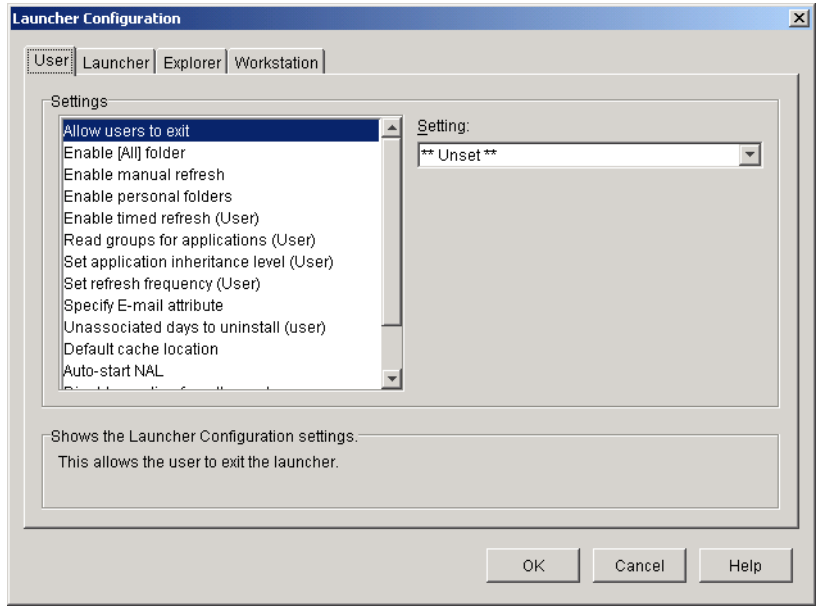
install applications from the cache, and uninstall applications. By default, this feature is disabled so as to maintain administrative control over these processes. However, you may find it necessary or desirable to enable this feature for some users:

To enable users to manage their applications:

1. In ConsoleOne, right-click the User, Group, or container object for the users whom you want to enable the Manage Applications feature > click Properties.
2. Click the Application Launcher tab to display the Launcher Configuration page.



3. Click Edit to display the Launcher Configuration dialog box.



4 In the Settings list (User tab), select Enable Manage Applications Dialog > set the option to Yes.

5 Click OK > OK to save the changes.

6 To use the feature, at a user's workstation, right-click the Application Launcher/Explorer icon > click Manage Applications.

For information about using the Manage Applications features, see the Help available in the Application Launcher/Explorer window.

8 Organizing Applications into Folders

ZENworks™ for Desktops Application Management enables you to organize Application objects into folders. Folders show up in the Novell® Application Launcher™/Explorer window and on the Windows* Start menu. Four types of folders are supported:

- **Application Folder (Administrator Created):** An application folder is created as an NDS® object. A single Application Folder object can consist of one folder or multiple folders (a multi-level folder structure). Once you create an Application Folder object, you can add any number of Application objects to the folder (or folders). Application Folder objects are the recommended folder type if you want to maintain consistent folders from user to user.
- **Custom Folder (Administrator Created):** A custom folder is created specifically for a single Application object in NDS. No other Application objects can be included in the folder. Custom folders support folders within folders, which means that you can create a custom folder structure. For example, while you could not have Calculator and Notepad in the same custom folder, you could create two subfolders within the same custom folder and place the two programs in the two subfolders (in other words, WINAPPS\CALCULATOR\CALC.EXE and WINAPPS\NOTEPAD\NOTEPAD.EXE).
- **Personal Folder (User Created):** A personal folder is created in Application Launcher/Explorer by individual users. Users can move Application objects from other folders into the personal folder. By default, the ability to create personal folders in disabled. For information about enabling this feature, see "Configuring Application Launcher/Explorer" on page 33.
- **System Folder (System Created):** A system folder is automatically created by Application Launcher/Explorer to store an Application object if the object has not been added to an application folder or a custom folder.

  If the Application object is associated with a user, the Application object's system folder is given the name specified in the User object's Full Name field (for example, John Smith). If the Application object is associated with another NDS object (Group, Workstation, container, and so forth), the system folder is given the name specified in the object's Description

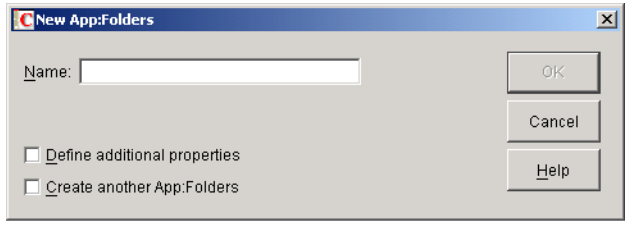
field (for example, System Test Group). If nothing is specified in the Full Name or Description field, the system folder is given the object's fully distinguished name. For example, an Application object associated with a user through his or her User object would be placed in a system folder that has the user's fully distinguished name, while an Application object associated with a user through a group would be placed in a folder that has the group's fully distinguished name.

The following sections provide instructions about creating an Application Folder object and adding Application objects to application folders or custom folders:

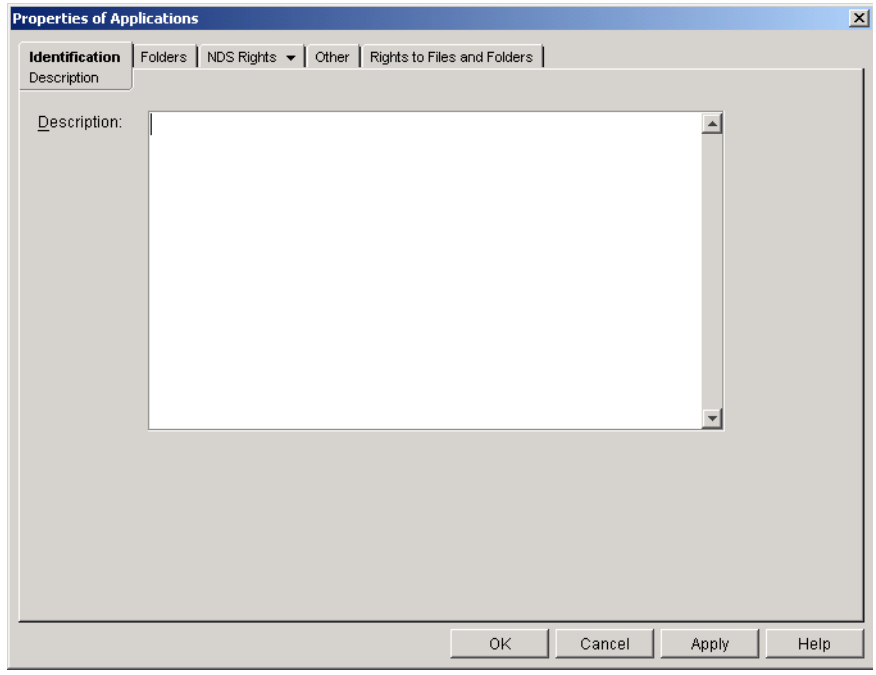
- "Creating Application Folder Objects" on page 90
- "Adding Applications to Application Folders" on page 92
- "Adding Applications to Custom Folders" on page 95

Creating Application Folder Objects

1. In ConsoleOne™, right-click the container in which you want to create the Application Folder object > click New > click Object to display the New Object dialog box.
2. Click App:Folders > click OK to display the New App:Folders dialog box.

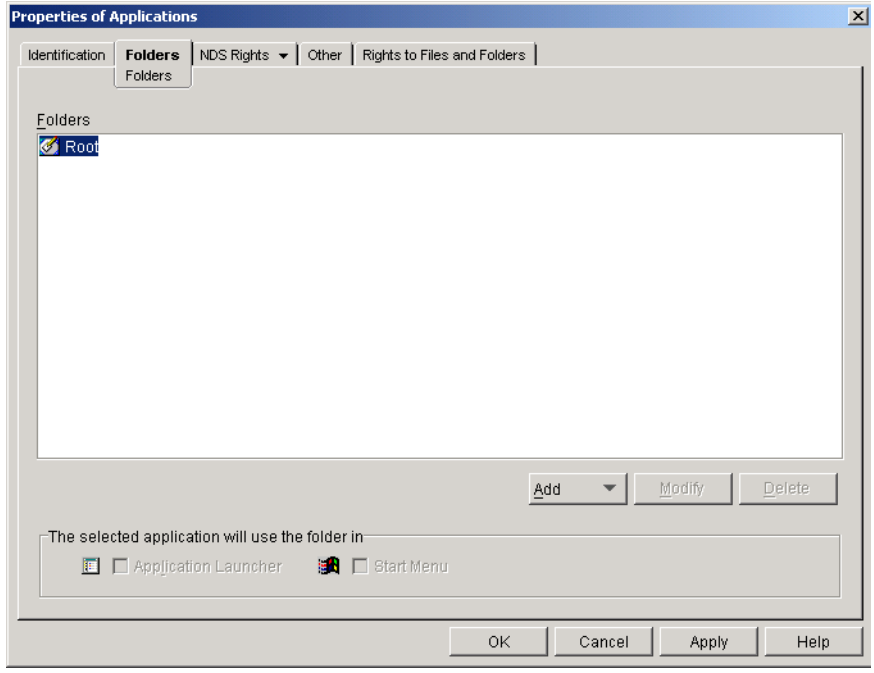


3. In the Name field, enter a name for the object. The name is displayed only in NDS and does not need to be the same name as the one you want to appear in the Application Launcher/Explorer window and on the Start menu on the users' workstations.
4. Check the Define Additional Properties box > click OK to display the Application Folder object's Description page.



5 If desired, enter a description in the Description box.

6 Click the Folders tab to display the Folders page.



The Folders tree contains one entry called Root. Root represents the Windows Start menu and the Application Launcher/Explorer window.

7 Use the Add, Modify, and Delete options to create the folder structure you want.

Add: To add a folder, select the folder (or Root) where you want to create a new folder > click Add > click Folder > type the name for the new folder > press Enter.

Modify: To change the name of a folder, select the folder in the Folders tree > click Modify > type the new folder name > press Enter.

Delete: To remove a folder, select the folder in the Folders tree > click Delete.

8 Click OK when you are finished creating the folder structure.

Adding Applications to Application Folders

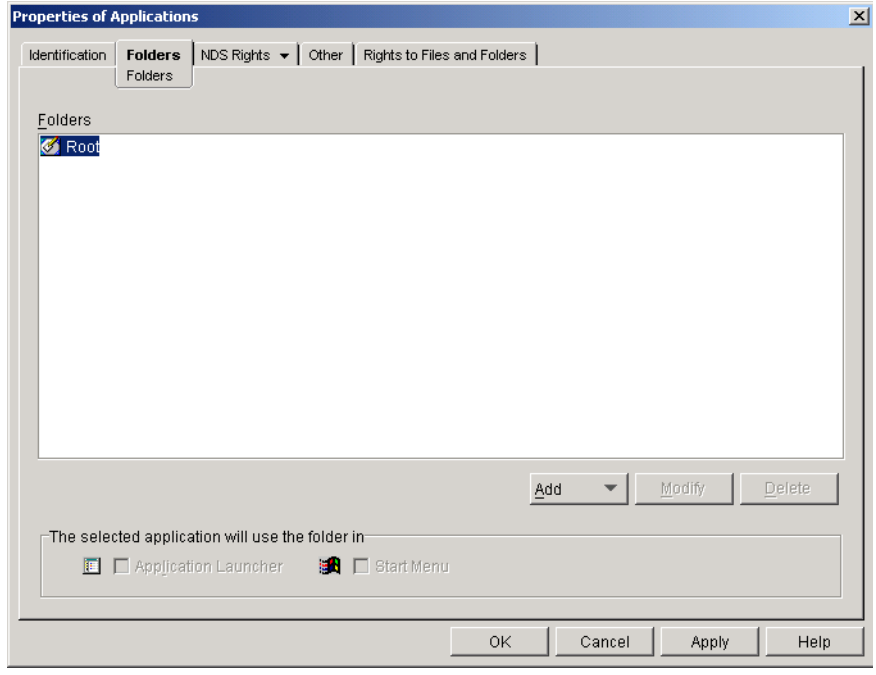
There are two ways you can add Application objects to application folders. You can add Application objects through the Folders page on the Application Folder object, or you can link individual Application objects through the

Folders page on the Application object. Using the Application Folder object enables you to add multiple Application objects to the application folder at one time. Using the Application object enables you to add the single Application object only.

- "Using the Application Folder Object to Add Applications" on page 93
- "Using the Application Object to Add the Application to a Folder" on page 94

Using the Application Folder Object to Add Applications

1. In ConsoleOne, right-click the Application Folder object you want to add the Application object to > click Properties.
2. Click the Folders tab to display the Folders page.



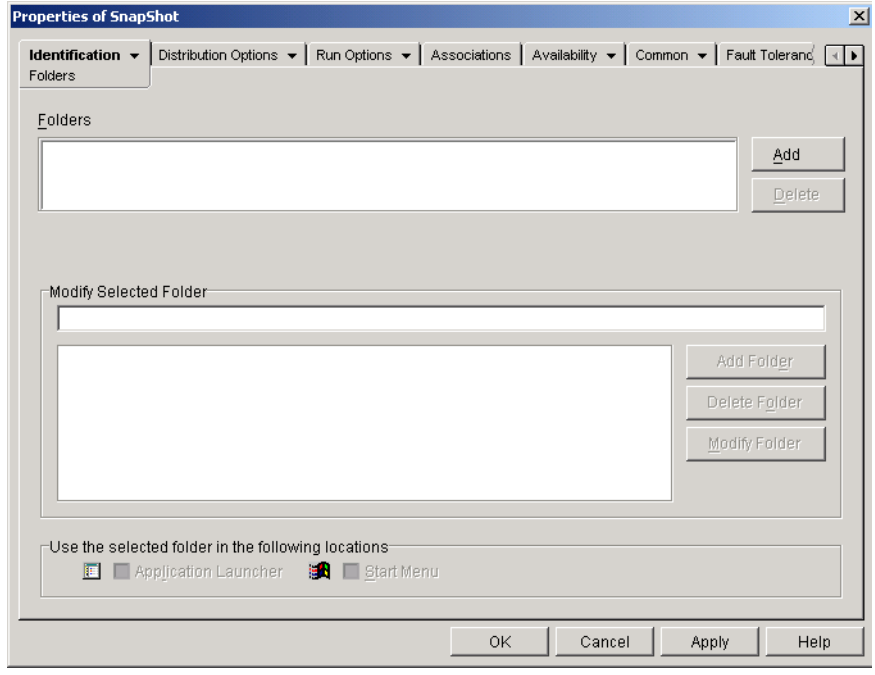
3. In the Folders tree, select the folder where you want to add the Application object.
4. Click Add > click Application > browse for and select the desired Application object > click OK to add it to the folder in the folders list.

5 When finished adding Application objects > click OK to save your changes.

Using the Application Object to Add the Application to a Folder

1 In ConsoleOne, right-click the Application object you want to add to a folder >click Properties.

2 Click the Identification tab > click Folders to display the Folders page.



3 Click the Add button > click Linked Folder to display the Select Object dialog box.

4 Browse for and select the Application Folder object you want to add the Application object to > click OK to display the Folder Object Structure dialog box.

5 Select the folder you want to add the Application object to > click OK to add it to the Folders list.

HINT: If you select the Root folder, the Application object will be displayed directly on the Start menu and at the root of the Application Launcher/Explorer window.

By default, the Application object will be displayed in the selected folder on the Start menu and the Application Launcher/Explorer window.

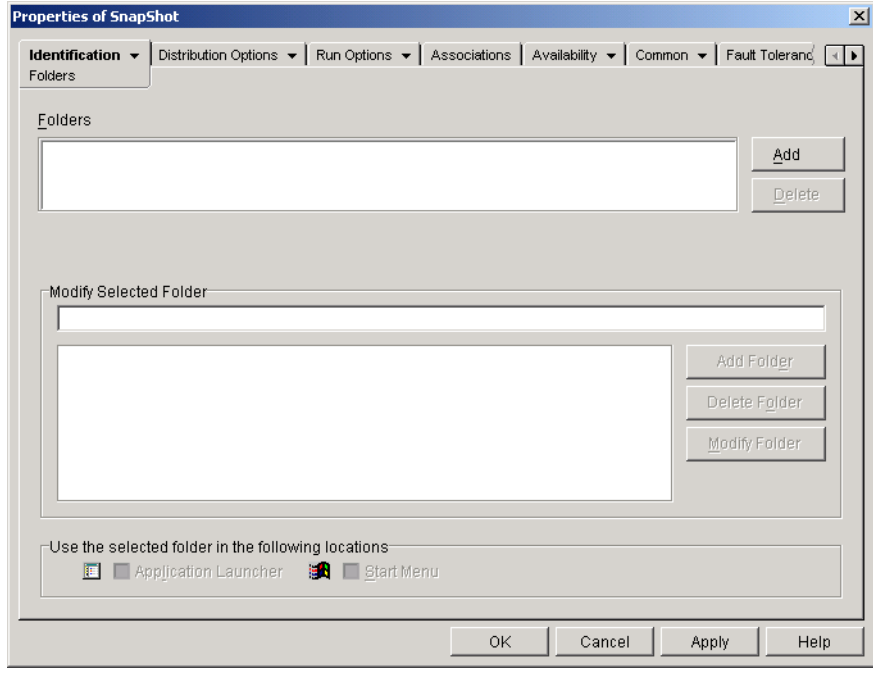
6. In the Use the Selected Folder in the Following Locations box, deselect Application Launcher if you don't want the Application object included in that folder in the Application Launcher/Explorer window and deselect Start Menu if you don't want the Application object included in that folder on the Windows Start menu.

   NOTE: A folder only displays in the Application Launcher/Explorer window and on the Start menu if it contains an Application object. If no Application objects have been linked to the folder, it is not displayed.

7. Click OK to save the changes.

Adding Applications to Custom Folders

1. In ConsoleOne, right-click the Application object you want to add to a folder > click Properties.
2. Click the Identification tab > click Folders to display the Folders page.



3 Click the Add button > click Custom Folder to add a New Folder entry to the Folders list.

4 In the Modify Selected Folder list, select the New Folder entry > click Modify Folder > type a name for the folder > press Enter.

5 If you want to add subfolders, select the folder > click Add Folder > type a name for the subfolder > press Enter.

IMPORTANT: If you add subfolders, the Application object is displayed in the lowest level folder.

6 Click OK to save the changes.

9 Reporting on Application Management Events

ZENworks™ for Desktops (ZfD) Application Management supports reporting on the success or failure of the following events:

- Launching of an application
- Distribution of an application
- Filtering of an application when the workstation does not meet the application's system requirements
- Removal of an application (uninstall)
- Caching of an application's installation source files

Events can be sent to a database, as SNMP traps to a management console, or recorded in a text log file. Because event reporting is established on a per application basis, you can use a database for reports from one application, SNMP traps for reports from a second application, and a text log file for reports from a third application. Or, you can use all three methods for different events for the same application (for example, launch events sent to a database and caching events sent to a log file).

The following sections provide information about setting up each type of event reporting and configuring Application objects to use one or more reporting methods:

- "Setting Up Database Reporting" on page 97
- "Setting Up SNMP Trap Reporting" on page 105
- "Setting Up Log File Reporting" on page 108
- "Configuring Applications to Use Reporting" on page 109
- "Generating Database Reports" on page 111
- "Understanding Log File Reports" on page 111

Setting Up Database Reporting

Novell® Application Launcher™/Explorer can record events to most ODBC-compatible databases, provided:

- Each user's workstation has the correct database driver installed.
- The connection is configured correctly in NDS®.

ZfD includes a Sybase* database that you can install. Sybase is also used for the Workstation Inventory database. If you plan to use a database for Application Management reports and you also plan to use Workstation Inventory, you can use the same database installation for both purposes.

NOTE: While the same database installation can be used for both Application Management and Workstation Inventory, each component still uses its own database file. Application Management creates a NAL.DB database file and Workstation Inventory creates a MGMTDB.DB database file.

To set up database reporting, you need to complete the following tasks:

- "Installing the Sybase Database" on page 98
- "Creating a ZENworks Database Object" on page 98
- "Configuring the Database's JDBC and ODBC Information" on page 100
- "Granting ODBC Property Rights to Users" on page 102
- "Enabling the ZENworks Database Policy" on page 103
- "Configuring Application Objects to Use Database Reporting" on page 105

Installing the Sybase Database

If you have not installed a database to use, you can install the Sybase database included on the *ZENworks for Desktops* product CD. For instructions, see Installing Application Management in Application Management in Deployment.

The Sybase database installation requires approximately 16 MB on a network server. However, as with all reporting databases, the database can expand rapidly to consume large amounts of disk space.

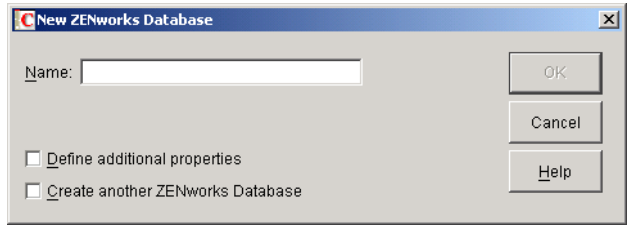
Creating a ZENworks Database Object

If you are using the Sybase database that comes with ZfD and have already installed it, the installation program creates the database object in NDS. You can skip to the next section, "Configuring the Database's JDBC and ODBC Information" on page 100.

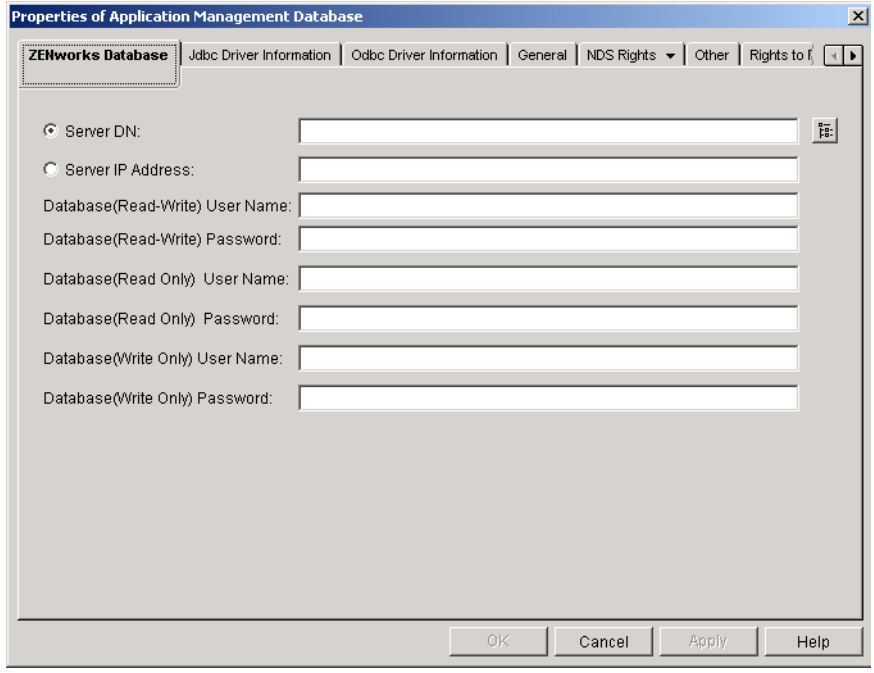
If you are using another database, such as the Oracle* database the comes with NetWare 5.1, you will need to create a ZENworks Database object in NDS to represent the database.

To create a ZENworks Database object:

1. Right-click the container in which you want to create the object > click New > click Object to display the New Object dialog box.
2. Select ZENworks Database > click OK to display the New ZENworks Database dialog box.



3. In the Name box, type a name for the database, such as ZfDAppManReports.
4. Select the Define Additional Properties box > click OK.



5 In the Server DN field, browse for and select the Server object for the server where the database is physically installed and running.

IMPORTANT: If the ZENworks database is located on a NetWare 4.*x* server, you need to enter the server's IP address in the Server IP Address field rather than adding the server's object to the Server DN field.

6 If you want to set up Read-Write, Read-Only, or Write-Only users, enter information in the appropriate user name and password fields.

7 Click OK to save the information.

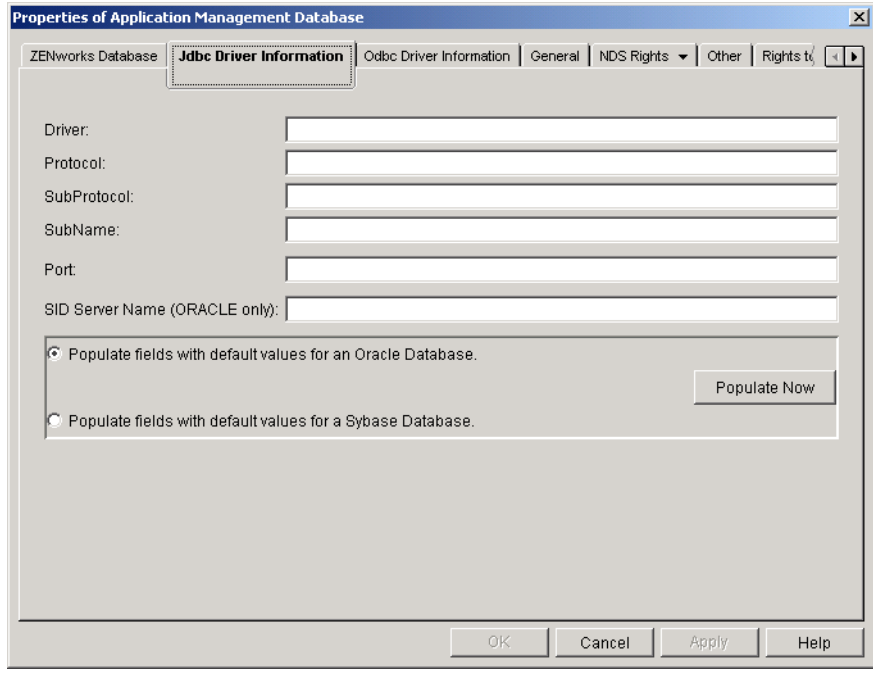
Configuring the Database's JDBC and ODBC Information

ConsoleOne™ uses a JDBC driver to pull information from the database for a set of predefined Application Management reports. If you want to use these predefined reports, you need to configure the database object with the correct JDBC driver information.

Application Launcher/Explorer uses an ODBC driver to write event information to the database. You need to configure the database object with the ODBC driver information that Application Launcher/Explorer will need.

To provide the JDBC and ODBC information:

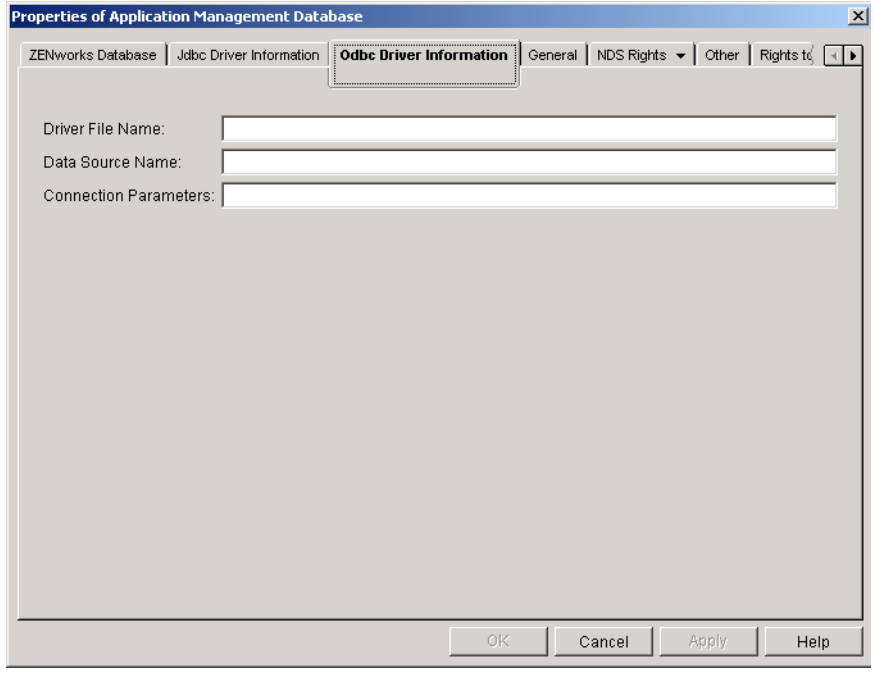
1. Right-click the database object > click Properties.
2. Click the JDBC Driver Information tab.



3. If you are using a Sybase or Oracle database, click the appropriate Populate option > click Populate Now to automatically populate the fields with the default information for a Sybase or Oracle database.

   or

   If you are using a different type of database, fill in the fields with the appropriate information for your database. Click the Help button for a description of each field.
4. Click the ODBC Driver Information tab.



5 Fill in the following fields:

Driver File Name: Enter the name of the ODBC driver file that resides on the workstation to enable Application Launcher/Explorer to access the database. For Sybase, the driver file name is dbodbc6.dll.

Data Source Name: Enter **`NAL Reporting`**.

Connection Parameters: Enter the appropriate commands to connect to the database. For Sybase, these commands are:

```
CommLinks=TCPIP{Host=IPAddress:2638};AutoStop=Yes;Integr
ated=No;DBN=NAL;ENG=IPAddress
```

where *IPAddress* is the actual IP address of the server where the database resides.

6 Click OK to save the JDBC and ODBC driver information.

Granting ODBC Property Rights to Users

You need to grant users Read and Compare rights to the ODBC properties you defined for the database object in the previous section. This allows

Application Launcher/Explorer to retrieve the ODBC information it needs to access the database.

To grant rights:

1. Right-click the ZENworks Database object > click Trustees of This Object.
2. Click Add Trustee.
3. Select [PUBLIC] > click OK to add [PUBLIC] to the list of trustees.

   Adding [PUBLIC] as a trustee gives every user Read and Compare rights to all of the database object's properties, including the various user names and passwords that can be used to access the database. To avoid this, you need to limit the [PUBLIC] access to the three ODBC properties.
4. In the Property list, select [All Attribute Rights] > click Delete Property to remove it from the list.
5. Click Add Property to display the Add Property dialog box > select zendbODBCConnectionParameters > click OK to add it to the Property list.

   The default rights, Read and Compare, are sufficient. You do not need to change these rights.
6. Repeat Step 5 to add the following two properties: zendbODBCDataSourceName and zendbODBCDriverFileName. Keep the default rights (Read and Compare).
7. Click OK > OK to save the changes.

Enabling the ZENworks Database Policy

Before Application Management can use the database, you need to:

- Activate a Database Policy in an NDS Service Location Package. The Database Policy simply points to the location of the ZENworks™ database you are using. A Service Location Package can have one Database Policy only, and each Database Policy can point to one database only. If you haven't created a Service Location Package, or if your current packages' Database Policies are being used for other databases, you will need to create a new Service Location Package. Instructions are provided in the steps below.
- Associate the Service Location Package with the containers where Application objects reside. This association is how Application

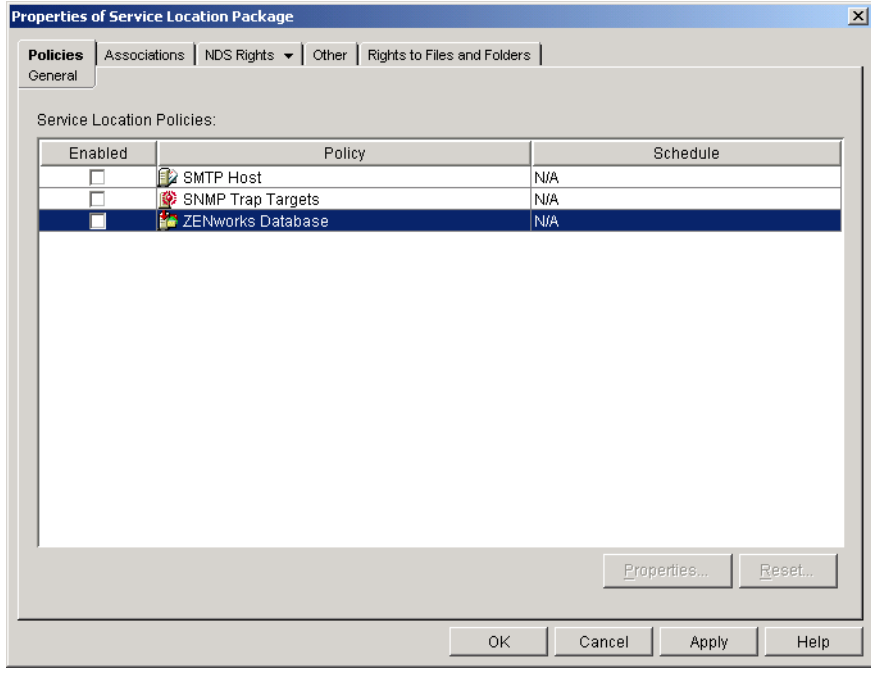
Launcher/Explorer knows which database to use when reporting events for the applications.

To do so, complete the following steps:

1. In ConsoleOne, right-click the Service Location Package you want to use > click Properties.

   or

   If you do not have a Service Location Package, right-click the container where you want to create one > click New > click Policy Package. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.



2. On the General page, check the box in the Enabled column to enable the ZENworks Database Policy.
3. Select the ZENworks Database Policy in the list > click Properties to display the Database Location page.
4. In the ZENworks Database DN field, browse for and select the database object you want to use for Application Management reporting. This

should be the same object you configured in "Configuring the Database's JDBC and ODBC Information" on page 100.

5. Click OK to return to the General page.
6. Click Associations to display the Associations page.

   You use this page to associate the Service Location Package with the containers where Application objects reside. If you have Application objects in multiple containers, make sure to add all containers to the list (unless you don't want events reported for applications in those containers, or want to use another database for those applications' events). If a parent container includes several containers where Application objects reside, you can select the parent container rather than each individual container.
7. Click Add > browse for and select the container you want to add > click OK to add it to the list.
8. Repeat Step 7 to add additional containers.
9. When you've finished adding containers, click OK to save the information.

Configuring Application Objects to Use Database Reporting

Once you've set up the database you will use for reporting, you can start configuring Application objects to use database reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use database reporting, SNMP trap reporting, or log file reporting, see "Configuring Applications to Use Reporting" on page 109.

Setting Up SNMP Trap Reporting

If you have a management console to collect SNMP traps, you can have Application Launcher/Explorer send SNMP traps to the management console.

To set up SNMP trap reporting, you need to complete the following tasks:

- "Enabling the SNMP Trap Targets Policy" on page 106
- "Configuring Application Objects to Use SNMP Trap Reporting" on page 108

Enabling the SNMP Trap Targets Policy

Before Application Management can use SNMP traps for event reporting, you need to:

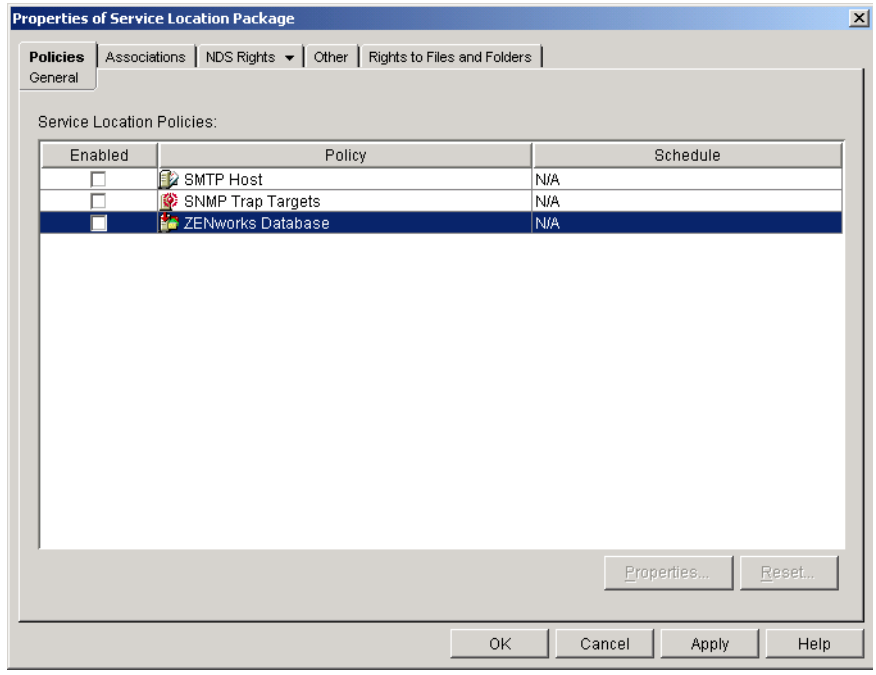
- Activate an SNMP Trap Targets Policy in an NDS Service Location Package. The SNMP Trap Targets Policy simply points to the IP address (or addresses) of the management console that will display the traps. A Service Location Package can have one SNMP Trap Targets Policy only. If you haven't created a Service Location Package, or if your current packages' SNMP Trap Targets Policies are being used for other databases, you will need to create a new Service Location Package. Instructions are provided in the steps below.
- Associate the Service Location Package with the containers where Application objects reside. This association is how Application Launcher/Explorer knows which database to use when reporting events for the applications.

To do so, complete the following steps:

1. In ConsoleOne, right-click the Service Location Package you want to use > click Properties.

   or

   If you do not have a Service Location Package, right-click the container where you want to create one > click New > click Policy Package. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.



2 On the General page, check the box in the Enabled column to enable the SNMP Traps Target Policy.

3 Select the policy in the list > click Properties to display the SNMP Trap Targets list.

4 Click Add to display the SNMP Target dialog box > enter the IP address of the workstation or server where the management console is running > click OK to add the IP address to the list.

5 Repeat Step 4 to add additional targets.

6 When you've finished adding targets, click OK to return to the General page.

7 Click Associations to display the Associations page.

You use this page to associate the Service Location Package with the containers where Application objects reside. If you have Application objects in multiple containers, make sure to add all containers to the list (unless you don't want events reported for applications in those containers, or want to use another management console for those applications' events). If a parent container includes several containers

where Application objects reside, you can select the parent container rather than each individual container.

8. Click Add > browse for and select the container you want to add >click OK to add it to the list.
9. Repeat Step 8 to add additional containers.
10. When you've finished adding containers, click OK to save the information.

Configuring Application Objects to Use SNMP Trap Reporting

Once you've enabled the SNMP Trap Targets Policy and have your management console running, you can start configuring Application objects to use SNMP trap reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use SNMP trap reporting, database reporting, or log file reporting, see "Configuring Applications to Use Reporting" on page 109.

Setting Up Log File Reporting

You can have Application Launcher/Explorer record events to a log file. This can be an individual log file located on the user's workstation or a common log file on a network server. When using a common log file, users must be given Read and Write rights to the log file, but Application Launcher/Explorer will automatically authenticate them to the log file location.

To set up log file reporting, you need to complete the following tasks:

- "Setting Up a Common Log File Location" on page 108
- "Configuring Application Objects to Use Log File Reporting" on page 109

Setting Up a Common Log File Location

With log file reporting, you have two options. You can have Application Launcher/Explorer log events for each individual user to a file on the user's local driver, or you can have Application Launcher/Explorer log events for all users to a file in a common network location.

If you want Application Launcher/Application Explorer to log events to a file in a common network location, you need to establish the network directory and provide users with Read and Write rights to files in the directory.

Because log file names are established on a per-application basis, you can have individual log files for each application (by specifying a different log file name for each Application object) or one log file for all applications (by specifying the same log file name for all Application objects). You cannot have log files on a per-user basis, unless you have Application Explorer/ Launcher save the files to the users' local drives.

Configuring Application Objects to Use Log File Reporting

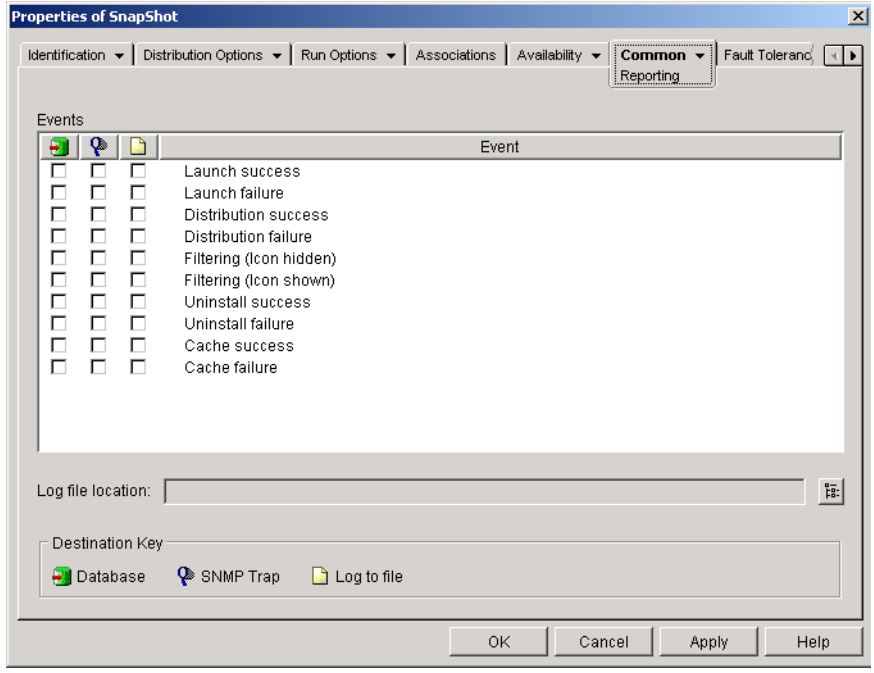
Once you've set up a common log file location (if you plan to use a common log file), you can start configuring Application objects to use log file reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use log file reporting, database reporting, or SNMP trap reporting, see "Configuring Applications to Use Reporting" on page 109.

Configuring Applications to Use Reporting

Application Launcher/Explorer must know which method of reporting (database, SNMP trap, or log file) to use for an application and which events to report for the application. You provide this information through the Application object's Reporting page (Common tab).

1. In ConsoleOne, right-click the Application object for which you want to configure reporting > click Properties.
2. Click the Common tab > click Reporting.



3 Fill in the following fields:

Events: Select the events (launch, distribution, filtering, uninstall, and cache) that you want reported and the destination where you want the events sent. You can send events to a database, to a management console via an SNMP trap, or to a text log file. If necessary, you can send events to multiple destinations (for example, both a database and a text log file).

Log File Location: If you selected a log file as the destination for any of the event reports, click the Browse button to display the Open dialog box > browse to and open the folder in which you want the log file located > type a name for the log file in the File Name box > click OK.

If you want to save the events for all users to the same log file, specify a log file in a common network location (one that users have Read and Write rights to). If you want users to have individual log files, specify a location on their local drive (for example, c:\temp).

4 Click OK to save the changes.

Generating Database Reports

When an event is recorded to a database, the following information is included: event type, date and time, user distinguished name, workstation distinguished name, workstation address, application distinguished name, application global unique identifier (GUID), application version string, and event strings.

You can use a database query to search for specific information. See your database documentation for information about performing queries.

In addition, Application Management includes a set of predefined reports for you to use. There are six basic reports that let you report on the success or failure of an event either by application, by user, or by workstation. For example, you could use the Failure by App report to view a list of all application distributions, sorted by user, that failed. Or you could use the Failure by User to view a list of all application launches, sorted by user, that failed.

To access the predefined reports:

1. Right-click the ZENworks Database object > click Reporting.
2. In the Available Reports list, expand the ZENworks NAL Reports category > select the report you want.
3. In the Event Type list, select the event you want to include in the report.
4. Click Run Selected Report.

Understanding Log File Reports

The following is a log file entry for a single event. Each field in the entry is described below.

```
"LaunchFail","11","5/10/2000:8:28PM",
"MAIN_TREE::.ssmith.xyzcorp",
"MAIN_TREE::ssmith_winnt.xyzcorp","01010480:00C04F6AABC6",
"MAIN_TREE::.testapp.org.xyzcorp",
"119C2580-949-11D3-BD13-00C04F6AABC6",
"5","-53429","","","","","","","524288"
```

Field	Example	Description
Event Type	LaunchFail	The event that occurred and whether it was successful or failed. Possible event types are: LaunchSuccess LaunchFail DistributionSuccess DistributionFail FilterShow FilterHide UninstallSuccess UninstallFail CacheSuccess CacheFail
Event Type Code	11	The code associated with the event. Possible codes are: 10 - LaunchSuccess 11 - LaunchFail 20 - DistributionSuccess 21 - DistributionFail 30 - FilterHide 40 - FilterShow 50 - UninstallSuccess 51 - UninstallFail 60 - CacheSuccess 61 - CacheFail
Date and Time	`5/10/2000:8:28PM`	The date (5/10/2000) and time (8:28PM) the event occurred.
User's Distinguished Name and Tree	MAIN_TREE::.ssmith.xyzcorp	The distinguished name and tree of the user for which the event occurred.

Field	Example	Description
Workstation Distinguished Name and Tree	MAIN_TREE::ssmith_winnt.xyzcorp	The distinguished name and tree of the workstation on which the event occurred. This field contains information only if the workstation has been imported into NDS as a Workstation object.
Workstation Address	01010480:00C04F6AABC6	The IPX™ or IP address of the workstation on which the event occurred.
Application Distinguished Name and Tree	MAIN_TREE::.testapp.org.xyzcorp	The distinguished name and tree of the Application object for which the event occurred.
Application GUID	119C2580-949-11D3-BD13-00C04F6AABC6	The global unique identifier assigned to the Application object. The GUID is located on the Application object's Options page (Distribution Options tab).
Application Version Number	5	The version number assigned to the Application object. Possible numbers range from 0 to 65535. The version number is located on the Application object's Options page (Distribution Options tab).
Event Code Major	-53429	The error code generated by Application Launcher/ Explorer. For error code explanations, see
Event Code Minor	no example	Additional error code.
Event String 1	no example	Error string.
Event String 2	no example	Error string.
Event String 3	no example	Error string.

Field	Example	Description
Event String 4	no example	Error string.
Event String 5	no example	Error string.
Application Flag	no example	Application object bitmask

10 Metering Software Licenses

ZENworks™ for Desktops (ZfD) Application Management integrates with Novell® Licensing Services (NLS) to enable you to track an application's usage and comply with the application's license agreement. When a user launches an application that has been configured as part of NLS, Novell Application Launcher™/Explorer checks to make sure that a license is available before running the application.

To set up software metering, complete the tasks in the following sections:

- Create a separate License container and one or more Metered Certificates for each application you want to track. For instructions, see the NLS documentation at the Novell Documentation Web site (http://www.novell.com/documentation).
- Configure the Application object in NDS® to use NLS and software metering. You won't be able to complete this step until after you've created an Application object for the application. For details about creating an Application object, see Distributing Applications in *Administration*. For details about configuring the Application object to use software metering, see Metering Software Licenses in *Administration*.

Installing Novell Licensing Services

Novell Licensing Services (NLS) must be installed before you can use Application Launcher/Explorer to meter software licenses. NLS is included with NetWare 4.*x*, 5, and 5.1, as well as NetWare Cluster Services. For information about installing NLS, see the documentation for the above listed products at the Novell Documentation Web site (http://www.novell.com/documentation).

Creating License Containers and Metered Certificates

You must create a License container for each application you want to meter. Within the License container, you then need to create one or more Metered Certificates that account for the number of licenses you have available for the application. For example, if you initially have 200 licenses for the application, you can create a Metered Certificate of 200 licenses. Later, if you purchase

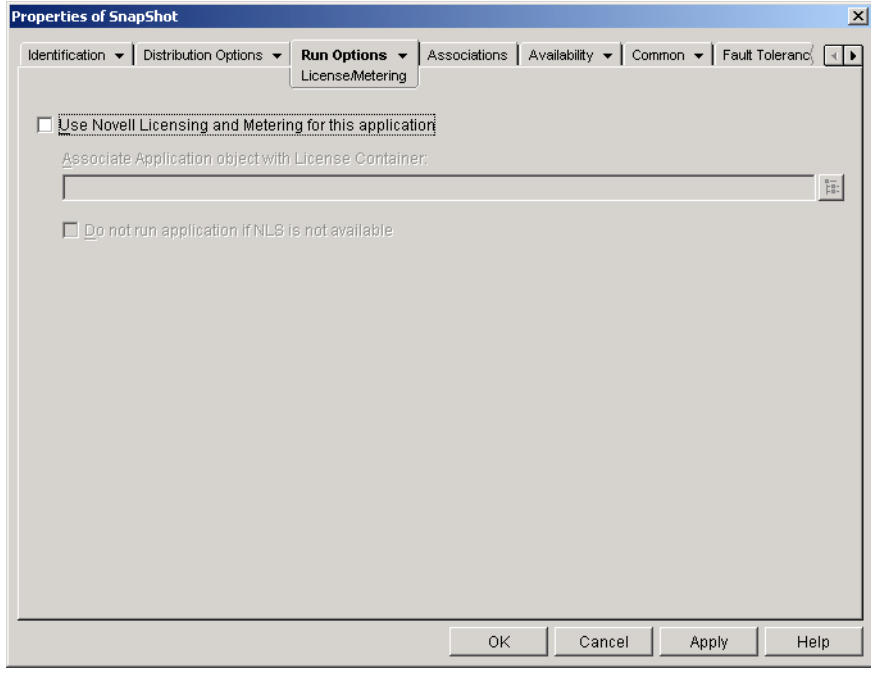
100 additional licenses, you can create a second Metered Certificate of 100 licenses. For instructions about creating License containers and Metered Certificates, see the NLS documentation at the Novell Documentation Web site (http://www.novell.com/documentation).

Configuring Applications to Use License Metering

Once you have installed NLS and have a License container and Metered Certificate created for an application, you need to configure the Application object to use NLS and license metering. This enables Application Launcher/Explorer to enforce the licensing you've established for the application.

To enable Application Launcher/Explorer to enforce an application's licensing:

1. In ConsoleOne, right-click the Application object > click Properties.
2. Click the Run Options tab > click License/Metering.



3. Check the Use Novell Licensing and Metering for this Application box to turn on the option.

4. In the Associate Application Object with License Container field, click the Browse button and select the application's License container.
5. If you don't want Application Launcher/Explorer to run the application when NLS is not available, check the Do Not Run if NLS Is Not Available box. Otherwise, Application Launcher/Explorer will run the application.
6. Click OK.

11 SnAppShot

To help you create installation packages for applications installed to Windows* 95/98 or Windows NT*/2000 workstations, ZENworks™ for Desktops (ZfD) Application Management provides the snAppShot™ utility. The following sections provide information about snAppShot and instructions for using the utility.

- "Understanding SnAppShot" on page 119
- "Preparing a SnAppShot Workstation" on page 120
- "Creating an Installation Package" on page 121

Understanding SnAppShot

SnAppshot records the changes that occur on a workstation as an application is installed. As the installation proceeds, snAppShot captures the differences between the workstation's pre-installation configuration state and the workstation's post-installation state, compares the two pictures, and creates an application installation package that consists of two Application object template (.AOT or .AXT) files, one or more application source (.FIL) files, and one file definition (FILEDEF.TXT) file.

Application Object Templates

You use the Application object template file to create the Application object in NDS®. Both Application object template (.AOT and .AXT) files contain the same information, which is used to populate the Application object property fields during creation of the object:

- The NDS name and workstation shortcut name to be given the Application object.
- The modifications that need to be made to the workstation's configuration settings (registry settings, INI settings, text file modifications, and so forth) during installation of the application.
- The macro definitions to use during installation.

- The list of application files to copy to the workstation during installation, including the source location to copy the files from and the target location to copy the files to.

The .AOT file is a binary file that cannot be edited; the .AXT file is a text file that can be modified with a text editor. If you need to modify the Application object template after snAppShot has created it, you should modify the .AXT file and use it to create the Application object. Otherwise, you should use the .AOT file because data from the .AOT file is imported more quickly.

Application Source Files

SnAppShot also tracks all of the application files that are copied to the workstation. These files, which become the application source files, are copied to a network source location, renamed numerically starting with 1, and given a .FIL file extension (for example, 1.FIL). Novell® Application Launcher™/Explorer uses these source files when installing the application to the workstation.

Application File Definition File

To map the .FIL files to their original files, SnAppShot creates a file definition (FILEDEF.TXT) file. This text file not only maps the .FIL files to the original files, but also specifies the target location and name to be used when installing the files to the workstation. For example:

```
1.fil=C:\DMI\WIN32\_DEISL1.ISU
2.fil=C:\DMI\WIN32\bin\Wdmiutil.dll
```

Preparing a SnAppShot Workstation

Before running snAppShot on a workstation to create an installation package for an application, you should:

- Make sure the workstation is clean. A clean workstation has only the operating system and the Novell Client™ installed.
- Make sure the workstation is representative of the type of workstation to which the application will be distributed. For example, if you are distributing the application to Windows 2000 users, run snAppShot on a Windows 2000 workstation, not a Windows 95 workstation.
  Depending on the application and what occurs during an install, it may be

necessary to create different Application objects to be used for different types of workstations.

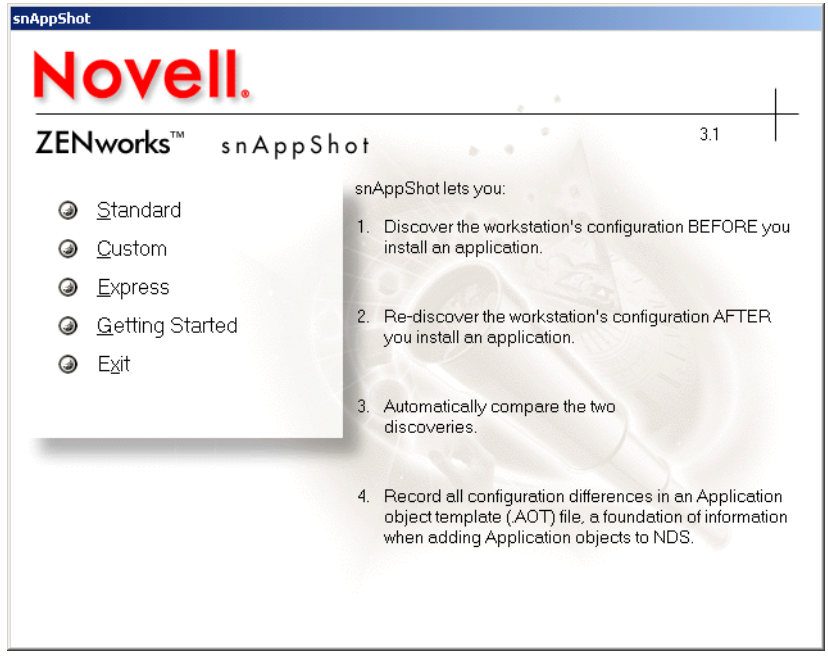
Creating an Installation Package

To create an installation package, snAppShot uses the following process:

- Takes an image of the workstation before the application is installed. You determine which drives are scanned and which configuration settings (registry settings, INI settings, and so forth) are recorded.
- Lets you install the application.
- Takes another image of the workstation, records the differences between the two images, and creates the installation package (.AOT and .AXT files, .FIL files, and FILEDEF.TXT file) in the network location you've specified.

To run snAppshot and create an installation package:

1. On the clean and representative workstation (see "Preparing a SnAppShot Workstation" on page 120), start snAppShot (SNAPSHOT.EXE) from the SYS:\PUBLIC\SNAPSHOT directory.


snAppShot
Novell
ZENworks™ snAppShot
3.1
Standard
Custom
Express
Getting Started
Exit
snAppShot lets you:
1. Discover the workstation's configuration BEFORE you install an application.
2. Re-discover the workstation's configuration AFTER you install an application.
3. Automatically compare the two discoveries.
4. Record all configuration differences in an Application object template (.AOT) file, a foundation of information when adding Application objects to NDS.

2 Select the mode you want to use:

Standard: Uses the snAppShot default preference file during the discovery process. In most cases, these preferences should be sufficient.

Custom: Allows you to choose a previously created preference file to apply to the discovery process or to use the snAppShot default preference file. Unlike the Standard mode, Custom mode lets you make changes to specific preferences to discover drives, files, folders, registry settings, and shortcuts.

Express: Lets you choose a previously created preference file. No modifications can be made to the preferences.

3 Follow the on-screen prompts to create the installation package. If you need more information than is provided on the screen, click the Help button.

12 Application Object Settings

An Application object includes many settings (properties) you can modify to manage the application. The following sections correspond to each tab on the Application object.

- “Identification Tab” on page 123
- “Distribution Options Tab” on page 132
- “Run Options Tab” on page 155
- “Associations Tab” on page 165
- “Availability Tab” on page 168
- “Common Tab” on page 189
- “Fault Tolerance Tab” on page 209

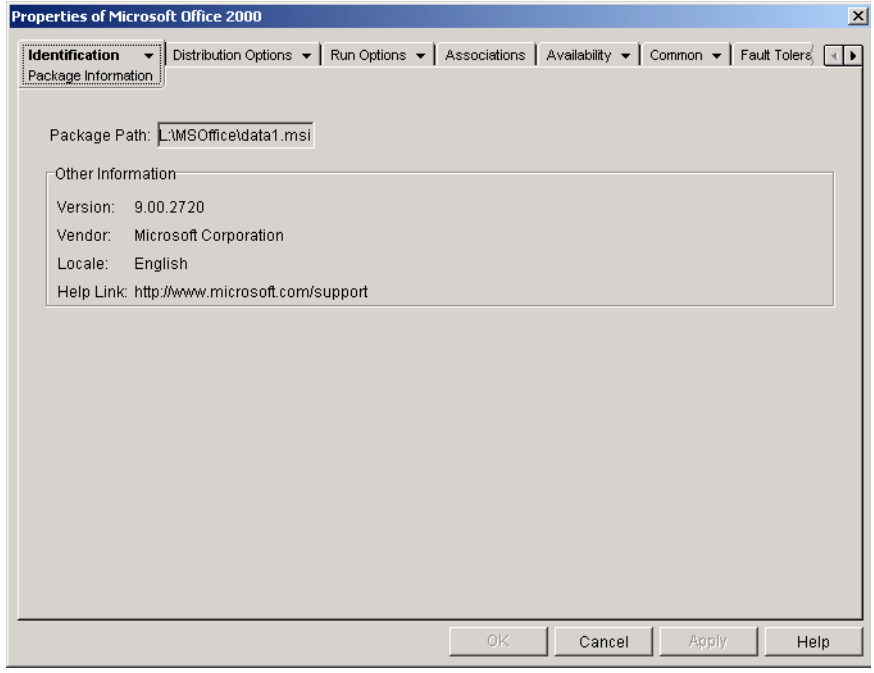
Identification Tab

The Identification tab includes the following pages to help you configure how the Application object is displayed to users:

- “Package Path” on page 124
- “Icon Page” on page 125
- “Description Page” on page 127
- “Folders Page” on page 128
- “Contacts Page” on page 130
- “Administrator Notes Page” on page 131

Package Information Page (.MSI Application Object Only)

The Package Information property page, shown below, displays information about the Microsoft* Windows* Installer package file (.MSI file) associated with the application. This page is for informational purposes only; you cannot use this page to modify the package's information.



Package Path

Displays the location of the .MSI file being used by the Application object.

Version

Displays the version of the .MSI file.

Vendor

Displays the creator of the .MSI file.

Locale

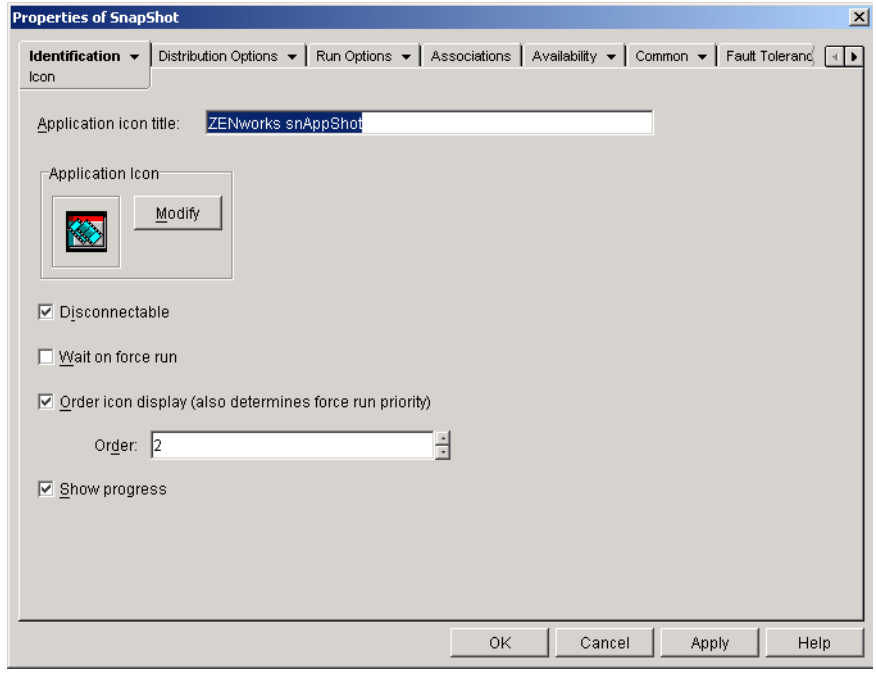
Displays the locale defined in the .MSI file.

Help Link

Displays the Web location to visit for information and help about the application.

Icon Page

The Icon property page, shown below, determines the Application object's icon that Novell® Application Launcher®/Explorer displays on a workstation. You can give the application icon a title, select the graphic to use for the icon, and give the icon a display order and force run priority. You can determine whether or not Application Launcher/Explorer should continue to display the Application object on the workstation when the user has disconnected from NDS®.



Application Icon Title

Enter the text you want to appear as the title for the Application object's icon wherever the icon is displayed. If necessary, you can use the Description page (Identification tab) for longer descriptions of the application.

Application Icon

Select the icon you want to appear wherever the Application object's icon is displayed. If you do not specify an icon, a default Application object icon is used.

Disconnectable

Check this option to mark the application as being able to be run on a workstation that is disconnected from NDS.

IMPORTANT: The application must be distributed to the workstation before the user can run it in disconnected mode, or the application must be cached on the workstation. You can force the application to be distributed on the workstation by selecting the Force Run characteristic (Associations page). You can force the application to be cached to the workstation by selecting the Force Cache characteristic (Associations page).

Wait on Force Run

This option applies only if the application and at least one other application are using the Force Run option (Associations page).

Check this option to force the application to wait until the application before it terminates. The application order is defined in the Order Icon Display (Also Determines Force Run Priority) field. Reboots will be queued until the final application has terminated.

Order Icon Display (Also Determines Force Run Priority)

This option performs two functions. First, it organizes the icons in Application Launcher/Explorer. Second, it dictates the order in which Application objects set as Force Run will run.

Ordering is left to right or top to bottom, depending upon the way the user views icons in Application Launcher/Explorer. You control ordering by entering a numeric value into the Order Icon Display text box. All Application objects you wish to order must have a numeric value. A value of zero gives the icon the highest priority. The maximum value is 999. If no order is specified, Application objects will be ordered alphabetically. When users who have been associated with these Application objects run Application Launcher/Explorer they will see a list of icons according to this order. If these applications have all been set to Force Run, Application Launcher/Explorer will run the applications in this order. Ordered and Force Run applications will run in sequential order without waiting for the last force run application to terminate, unless the Wait on Force Run option is enabled.

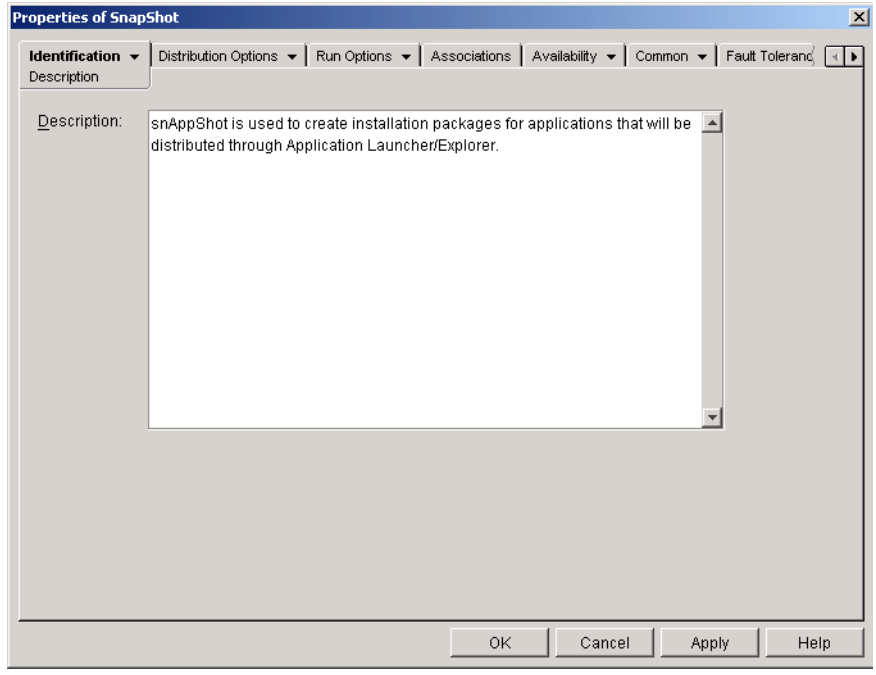
Show Progress

This option displays a progress bar to users anytime an application is distributed to or removed from their workstations. Turn off this option if you

are distributing only a small change, such as a registry modification. Turn on this option if you are distributing or removing a large application and want to give the user an idea of how long the process will take.

Description Page

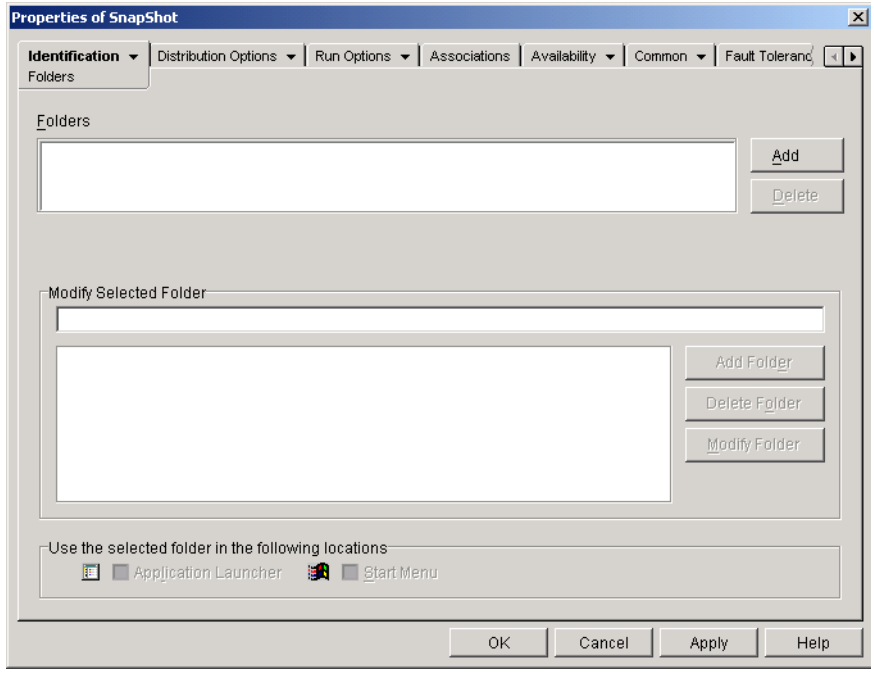
The Description property page, shown below, provides users more complete information about the Application object than the application icon title allows.



If you have enabled the Prompt User Before Distribution option (Distribution Options > Options page), users will see this description when Application Launcher/Explorer first distributes the application to them. They can also view an Application object's properties to see the description. To display the properties, right-click the Application object (on the workstation) > click Properties.

Folders Page

The Folders property page, shown below, lets you specify the folders where you want Application Launcher/Explorer to place the application when distributing it to a workstation.



You can add the application to two different types of folders: a custom folder and a linked folder.

A custom folder is a folder created specifically for the Application object. No other Application objects can be included in the folder. Custom folders support folders within folders, which means that you can create a custom folder structure. For example, although you could not have Calculator and Notepad in the same custom folder, you could create two subfolders within the same custom folder and place the two programs in the two subfolders (in otherwords, WINAPPS\CALCULATOR\CALC.EXE and WINAPPS\NOTEPAD\NOTEPAD.EXE).

A linked folder is simply an association to an existing Application Folder object. The Application Folder object must already exist in NDS. If the

Application Folder object includes multiple folders (a folder structure) you can add the application to any folder in the structure.

HINT: If you plan to create a complex folder structure for the applications you distribute, we recommend that you use an Application Folder object and then link Application objects to the Application Folder object. An Application Folder object requires you to define the folder structure one time only, whereas custom folders must be defined for each Application object. If you choose to use custom folders for your folder structure, make sure you use the same folder names when defining the custom folder structure for each Application object. Any variation will cause Application Launcher/Explorer to create different custom folder structures.

Folders

The Folders list displays the custom folders and linked folders (Application Folder objects) that the application has been added to. Application Launcher/Explorer will create (if necessary) the listed folders when distributing the application to the workstation.

Add

Click Add to add a custom folder or a linked folder to the Folders list. If you want the application to appear in multiple folders, add each folder to the list.

When you add a custom folder, it appears in the list with New Folder as its title. You can select the folder in the list and use the Modify Selected Folder box to change the folder's name or to add subfolders to it (see Modify Selected Folders below).

Delete

Select a folder in the folder list > click Delete to remove the folder from the list.

Modify Selected Folder

The Modify Selected Folder box lets you modify name and structure information for the folder that is selected in the Folders list. You can modify information for custom folders and linked folders. However, once you modify a linked folder's information, it is converted to a custom folder.

Add Folder

Select the folder in the folder tree > click Add to add a subfolder to the folder. After you add a subfolder and Application Launcher/Explorer refreshes, users will see the application in the subfolder rather than in the folder.

Delete Folder

Select the folder in the folder tree > click Delete to delete the folder.

Modify Folder

Select the folder in the folder tree > click Modify to change the name of the folder.

If you want to add the application to a new subfolder of an existing folder, enter *existing_folder_name\new_subfolder_name*.

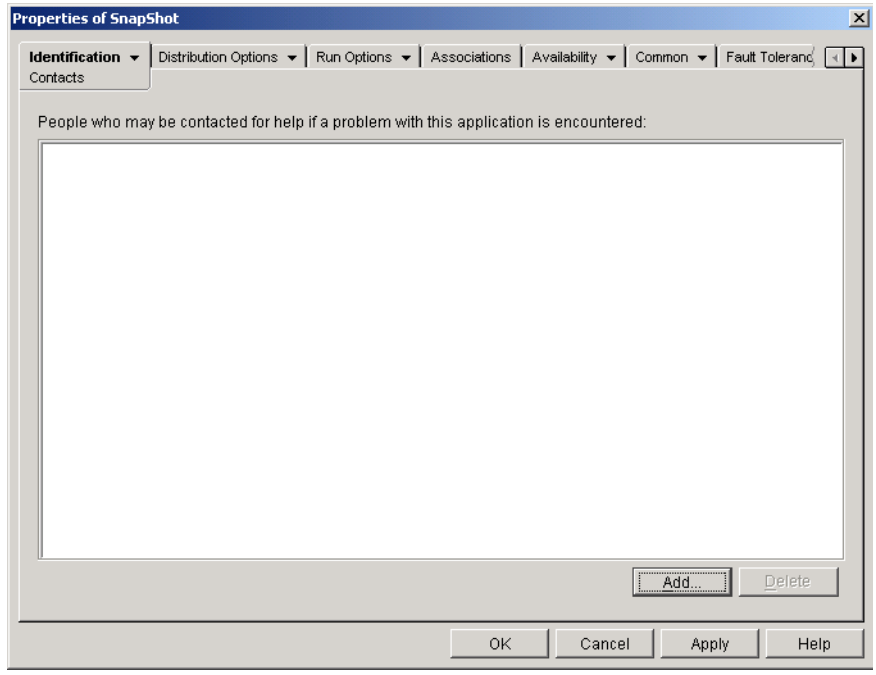
Use the Selected Folder in the Following Locations

Application Launcher/Explorer can display folders on the Windows Start menu and in the Application Launcher/Explorer window (provided these locations are enabled on the Application object's Associations page). Select a folder in the Folders list > check the boxes for the locations where you want to use the folder.

If you don't select either location, Application Launcher/Explorer will still display the Application object on the Start menu and in the Application Launcher/Explorer window (if these locations are enabled on the Associations page), but the object will not appear in the folders you've defined.

Contacts Page

The Contacts property page, shown below, lists the names, e-mail addresses, and phone numbers of the application's support staff. Users can access this information through the Application object's properties. To display the properties, right-click the Application object (on the workstation) > click Properties > click Help Contacts.

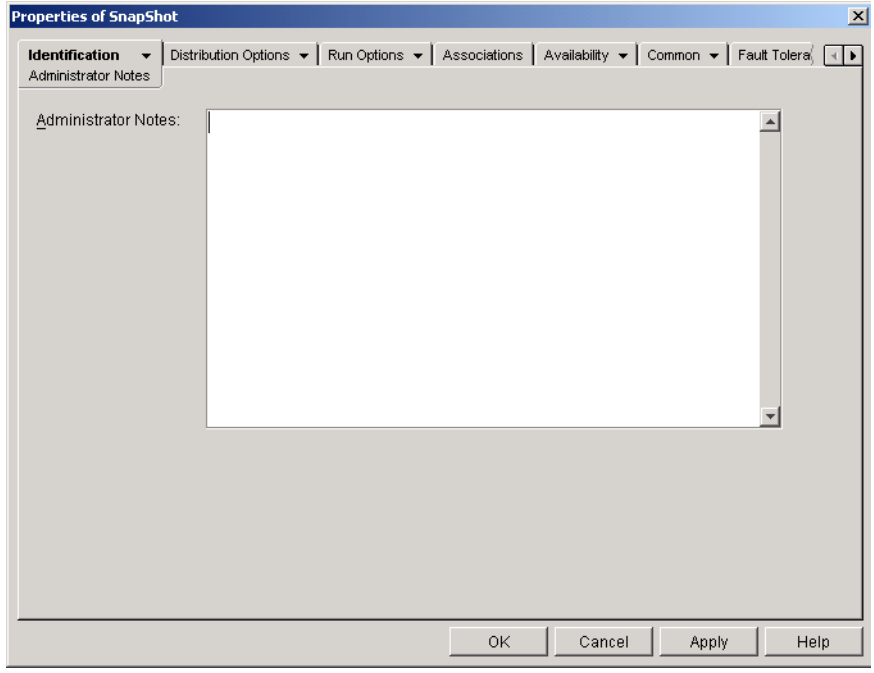


You can tailor the information to direct users to the support staff at their locations. If you enter the support contact's e-mail address, users will be able to send an e-mail message directly from the Help Contacts page of the Properties dialog box.

NOTE: Users must have the NDS rights required to read the E-Mail Address (MailboxID) and Telephone (Telephone Number) attributes of the users defined as contacts.

Administrator Notes Page

The Administrator Notes property page, shown below, lets you record notes for yourself or other administrators. For example, you could remind yourself about special settings for an application. Or, if you have several administrators, you could write a history of upgrades and file changes.



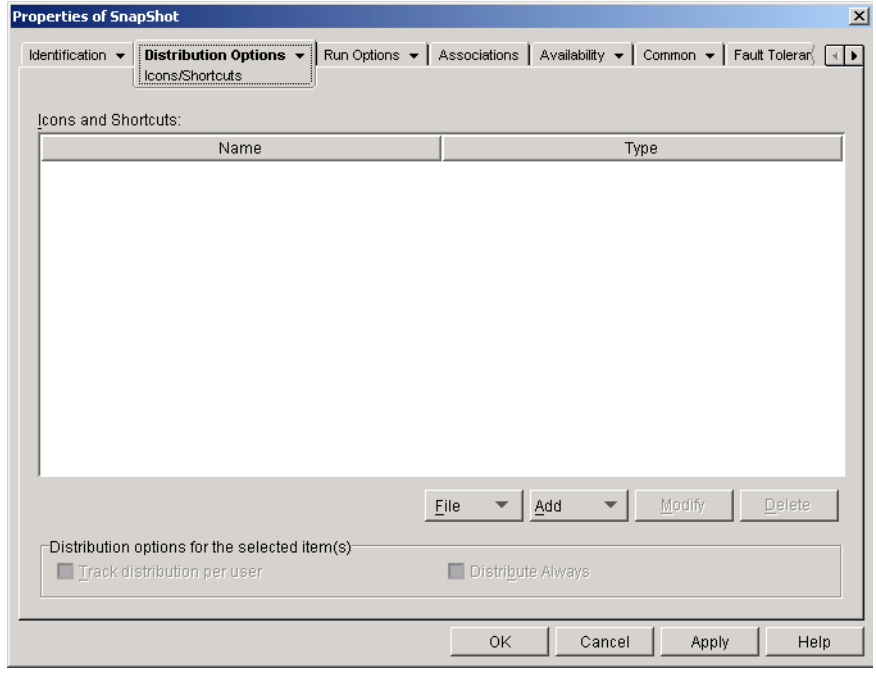
Distribution Options Tab

The Distribution Options tab includes the following pages to help you configure how the Application object is distributed to users:

- “Icons/Shortcuts Page (.AOT/.AXT Application Object Only)” on page 133
- “Registry Page” on page 135
- “Application Files Page (.AOT/.AXT Application Object Only)” on page 137
- “INI Settings Page” on page 141
- “Text Files Page (.AOT/.AXT Application Object Only)” on page 145
- “Distribution Scripts Page” on page 147
- “Pre-Install Schedule Page (.AOT/.AXT Application Object Only)” on page 149
- “Options Page” on page 152

Icons/Shortcuts Page (.AOT/.AXT Application Object Only)

The Icons/Shortcuts property page, shown below, determines the icons and shortcuts that Application Launcher/Explorer will create when distributing the application to the workstation. You can add the application's icon as an item in a program group or as a shortcut on the workstation's desktop or in a folder. You can also delete existing icons, shortcuts, and program groups.



The icons and shortcuts you add with this page are in addition to the Application object's icon. While the Application object's icon may cause various actions to occur, including installing the application or running it, the icons and shortcuts defined on this page link directly to the application's executable file and simply launch the application.

You can use icons and shortcuts in combination with other options to create the user environment you want. For example, you could define the icons and shortcuts you want created and configure the Application object to run one time (Run Options > Applications). When a user selected the Application object, Application Launcher/Explorer would run the application one time, create the icons and shortcuts, perform any other tasks specified by the Application object's properties, and then remove the Application object's icon

from the workstation. Thereafter, the user would need to select the icon or shortcut to launch the application.

IMPORTANT: Program groups and program group items are supported on Windows 95/98 workstations but not on Windows NT/2000 workstations. Shortcuts are supported on all four Windows versions.

Icons and Shortcuts

This list displays the icons and shortcuts that will be created when the application is distributed to a workstation.

File

Click File > Find to search for icon and shortcut definitions that include certain information.

Click File > Import to import icons and shortcuts from another Application object.

Add

Click Add to add a new program group, program group item, or shortcut.

Modify

Select an icon or shortcut from the Icons and Shortcuts list > click Modify to change the information associated with it.

Delete

Select an icon or shortcut from the Icons and Shortcuts list > click Delete to delete it from the list.

Track Distribution Per User

Use this option if you have enabled users with roaming profiles on your network and you need to track a particular icon and shortcut definition.

In the Icons and Shortcuts list, select the icon or shortcut you want to track > check this option.

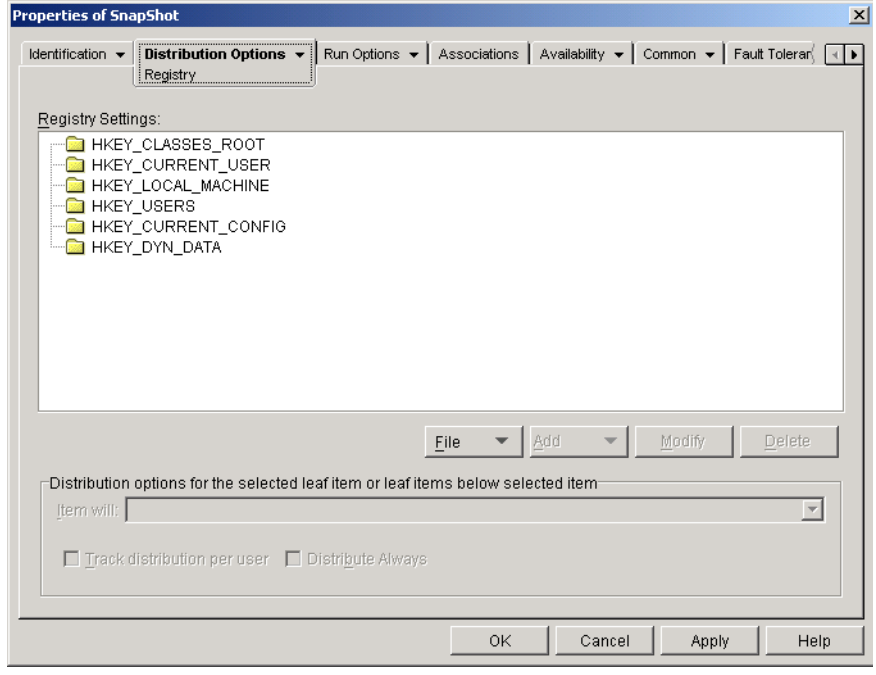
Distribute Always

Use this option if you want to create or delete the icons and shortcuts every time the application is distributed.

In the Icons and Shortcuts list, select the icon or shortcut to always distribute > check this option.

Registry Page

The Registry property page, shown below, determines the registry settings that Application Launcher/Explorer will create or delete when distributing the application to a workstation.



Registry Settings

The Registry Settings tree displays all settings that have been defined for the registry. When the application is distributed to a workstation, Application Launcher/Explorer will modify the workstation's registry according to the settings in this tree.

If you used a .AOT, .AXT, or .MSI file when creating the Application object, the tree automatically includes all registry settings that are defined in those templates.

File

Click File > Find to search for specific keys, value names, or value data in the registry.

Click File > Import to import registry settings from another Application object's .AOT or .AXT file, or from a registry file (.REG).

Click File > Export to export the registry settings to a registry file (.REG). To export the settings to a .AOT or .AXT file format, you must export the entire Application object using the Export Application Object option located on the Tools > Application Launcher Tools menu.

Add

This option lets you add registry settings to the Registry Settings tree. Only settings displayed in the Registry Settings tree will be created or deleted when the application is distributed.

To add a registry key or value, select the registry folder where you want to add the key, or select the key where you want to add a value > click the Add button > choose one of the following options:

- **Key:** Adds a key to the selected registry folder.
- **Binary:** Adds a binary value to the selected key.
- **Expand String:** Adds an expand string value to the selected key.
- **Default:** Adds a default string value to the selected key.
- **DWORD:** Adds a DWORD value to the selected key.
- **Multi-String Value:** Adds a multi-value string to the selected key.
- **String:** Adds a string value to the selected key.

Once the key or value is added to the Registry Settings tree, you can use the Distribution Options list to determine whether or not the key or value is created in the workstation's registry or deleted from the registry.

You can use a macro for a key name, value name, or value data. For information about macros, see Chapter 13, "Macros," on page 217.

Modify

Select the key or value you want to modify > click Modify.

Delete

Select the key or value you want to delete > click Delete. When you delete a key, everything subordinate to the key will also be deleted.

Distribution Options

Use these options to set individual distribution options for settings included in the Registry Settings tree.

Item Will

Use this option to specify whether the registry setting will be created or deleted during distribution.

In the Registry Settings tree, select the setting > select either Create or Delete from the Item Will list. If you delete a setting that has subordinate settings, the subordinate settings will also be deleted.

Track Distribution Per User

Use this option if you have enabled users with roaming profiles on your network and you need to track a particular registry setting.

In the Registry Settings tree, select the registry setting you want to track > check Track Distribution Per User.

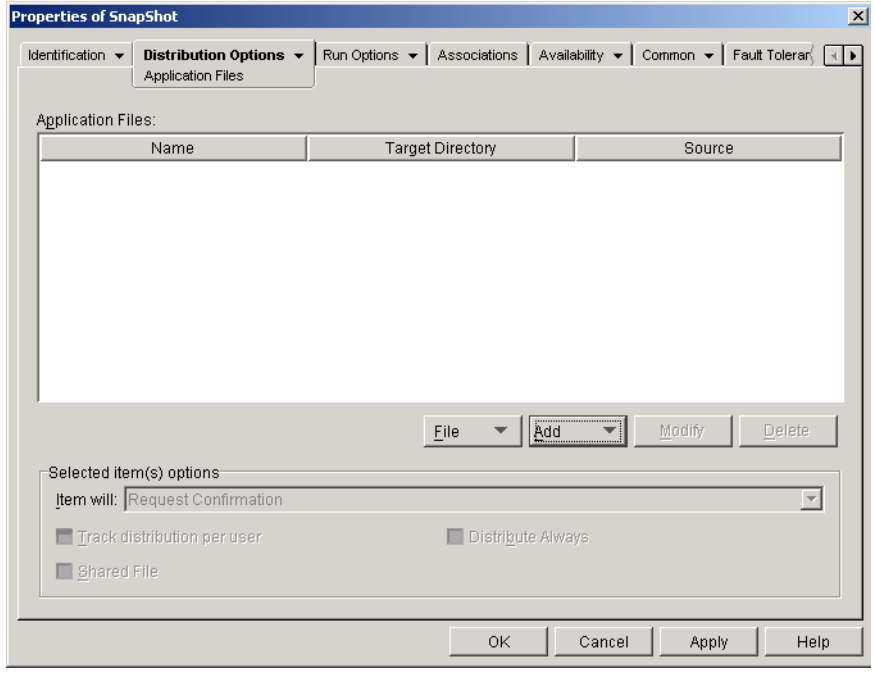
Distribute Always

Use this option if you want to create or delete a registry setting every time the application is distributed.

In the Registry Settings tree, select the registry setting to always distribute > check Distribute Always.

Application Files Page (.AOT/.AXT Application Object Only)

The Application Files property page, shown below, specifies the application files that Application Launcher/Explorer will install or remove when distributing the application to a workstation.



Application Files

The Application Files list displays all files and directories that will be installed or removed during distribution. The name, target directory (the location on the workstation where the file will be installed), and source (the file or directory that is being used to install the file) are listed for each application file or directory.

If you used a .AOT or .AXT file when creating the Application object, the list automatically includes all files and directories that are defined in those templates.

File

Click File > Find to search for specific items in the list. You can search for text in the Name, Target Directory, or Source fields. For example, you could search for all files and directories that are being installed under the Program Files directory.

Click File > Import to import application files and directories from another Application object's .AOT or .AXT file.

Add

This option lets you add files or directories to the Application Files list. Only files and directories displayed in the Application Files list will be installed to or removed from the workstation during distribution.

- **File:** To add a file to the list, click Add > File to display the Edit Files dialog box.

  In the Source File field, specify the file to be used as the source of the installation. You can use a mapped drive, UNC path, or macro, or you can browse and select the file (for example, \\SERVER1\VOL1\BOOKMARKS\BOOKMARK.HTM or %SOURCE_PATH%\BOOKMARK.HTM). The source file can be a single file that will be copied, multiple files (for example, %SOURCE_PATH%*.*), or a snAppShot™ application source (.FIL) file.

  If you are deleting a file from the workstation, leave this field blank.

  In the Target File field, specify the file on the workstation where the source file will be copied (for example, C:\PROGRAM FILES\NOVELL\BROWSER\BOOKMARK.HTM). You can also substitute a macro for the target path (for example, %TARGET_PATH%\BOOKMARK.HTM). If you are copying multiple files using a wildcard (*.*), specify the destination directory only (for example, C:\PROGRAM FILES\NOVELL\BROWSER\). If you are deleting the file from a workstation, enter the full path for the file from the perspective of the workstation > check the Target File to Be Deleted box.

- **Directory:** To add a directory to the list, click Add > Directory to display the Edit Directory dialog box.

  In the Name field, specify the directory to create or delete on the workstation (for example, C:\PROGRAM FILES\NOVELL). You can also substitute a macro for the directory path (for example, %DIRECTORY_TARGET_PATH%\NOVELL).

  Check the Directory to Be Deleted box if you want to delete the directory from the workstation. Otherwise, the directory will be created on the workstation.

Modify

Select the file or folder you want to modify > click Modify.

Delete

Select the file or folder you want to delete > click Delete to remove it from the Application Files list.

Selected Item(s) Options

Use these options to set individual distribution options for files and directories included in the Application Files list.

Item Will

Select a file in the Application Files list > select one of the following options from the Item Will list:

- **Copy Always:** Copies the file regardless of whether the file currently exists on the workstation.
- **Copy if Exists:** Copies the file only if the file currently exists on the workstation.
- **Copy if Does Not Exist:** Copies the file only if the file does not currently exist on the workstation.
- **Copy if Newer:** Copies the file only if its date and time are newer than the existing file's date and time, or if the file does not currently exist on the workstation.
- **Copy if Newer and Exists:** Copies the file only if it already exists on the workstation and has an older date or time.
- **Copy if Newer Version:** Copies the file only if its internal version is newer than the existing file's version (if version information is present). This is useful if you want to update the version of a .EXE or .DLL based on the compiled version information.
- **Request Confirmation:** Prompts the user to verify that the file should be copied.
- **Copy if Different:** Copies the file if its date, time, or size is different than the existing file's date, time, or size.
- **Delete:** Deletes the file from the workstation.

Select a folder in the Application Files list > select one of the following options from the Item Will list:

- **Create:** Creates the directory on the workstation.

- **Delete:** Deletes the directory from the workstation.

Track Distribution Per User

Use this option if you have enabled users with roaming profiles on your network and you need to track a particular file or directory on a per-user per-workstation basis.

In the Application Files list, select the file or folder you want to track > check Track Distribution Per User.

Distribute Always

Use this option if you want to create or delete a file or folder every time the application is distributed.

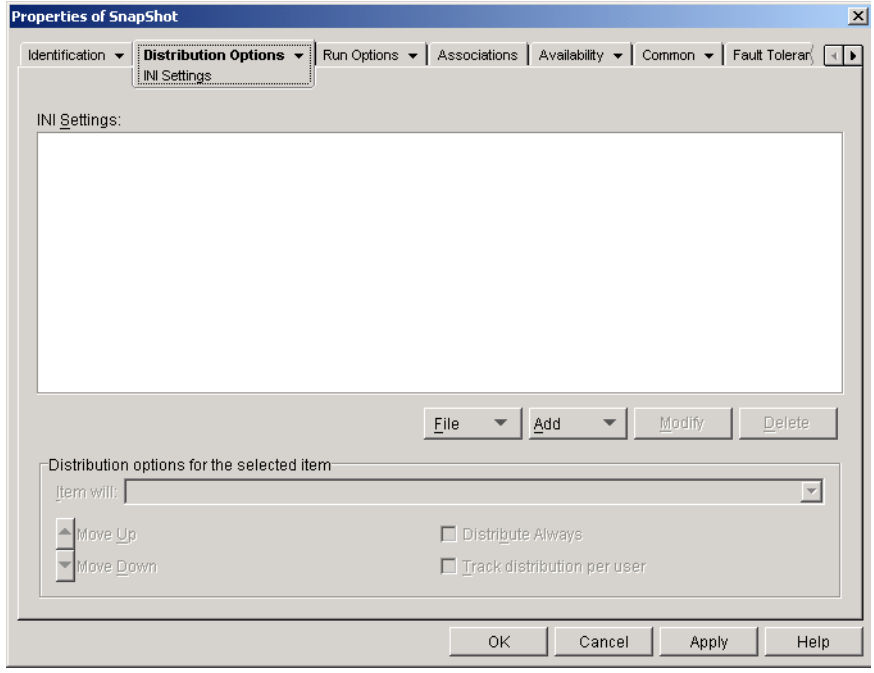
In the Application Files list, select the file or folder to always distribute > check Distribute Always.

Shared File

Use this option to mark a file as a shared file (that is, one that will be used by more than one application). Shared files are usually Windows .DLL files. SnAppShot detects shared files when it discovers application installation changes on a workstation.

INI Settings Page

The INI Settings property page, shown below, determines the .INI settings that Application Launcher/Explorer will create or delete when distributing the application to a workstation.



INI Settings

The INI Settings tree displays the .INI files that have been defined for the application, as well as all sections and values that have been added to each .INI file. When the application is distributed to a workstation, the .INI files are created if necessary, or existing .INI files are modified as needed.

If you used a .AOT, .AXT, or .MSI file when creating the Application object, the tree automatically includes all .INI settings that are defined in those templates.

File

Click File > Find to search for specific files, sections, or values.

Click File > Import to import .INI settings from another Application object's .AOT or .AXT file, or from a .INI file.

Click File > Export to export the settings to a .INI file. To export the settings to a .AOT or .AXT file, you must export the entire Application object using the Export Application Object option located on the Tools > Application Launcher Tools menu.

Add

This option lets you add INI settings to the INI Settings tree. Only settings displayed in the INI Settings tree will be created or deleted when the application is distributed. You can add a file to the tree, a section to a file, or a value to a section.

To do so, select the appropriate item in the tree > click the Add button > choose one of the following options:

- **File:** Adds a file to the INI Settings tree. After you name the file, you can begin adding sections to it.
- **Section:** Adds a section to the selected file. After you name the section, you can begin adding values to it.
- **Value:** Adds a value to the selected section. You must specify the value name and value data.

Once you've added a value to the INI Settings tree, you can use the Distribution Options list to determine whether or not it is created or deleted from the workstation. If the value needs to be created but the file or section does not exist, Application Launcher/Explorer will create the file or section before adding the value.

You can use a macro for a section name, value name, or value data.

Modify

Select the file, section, or value you want to modify > click Modify.

Delete

Select the file, section, or value you want to delete from the INI Settings tree > click Delete. When deleting a file or section, everything subordinate to it will also be deleted.

Distribution Options

Use these options to set individual distribution options for value settings included in the INI Settings tree.

Item Will

Select a value in the INI Settings tree > select one of the following options from the Item Will list:

- **Create Always:** Creates the value regardless of whether the value currently exists in the section.
- **Create if Does Not Exist:** Creates the value only if the value does not currently exist in the section.
- **Create if Exists:** Creates the value only if the value currently exists in the section.
- **Create or Add to Existing Section:** Creates the value if the value does not currently exist in the section. If the value exists, it adds this value to the section in addition to the one that already exists. This is useful, for example, if you need multiple values of the same type, such as two "DEVICE=" values.
- **Create or Append to Existing Value:** Creates the value if the value does not currently exist in the section. If the value exists, it appends the data for the new value to the existing value. The first character in the value data needs to be the separator character, such as a space.
- **Delete:** Deletes the value from the section.
- **Delete or Remove From Existing Value:** Deletes the value from the section, or, if the value has multiple data entries, removes this value's data entry from the value. For example, suppose the following setting is in the WIN.INI file: `Run = sol.exe calc.exe`. Using this option, you can remove just calc.exe, leaving the following: `Run = sol.exe`. The first character in the value data needs to be a separator character, such as a space.

Track Distribution Per User

Use this option if you have enabled users with roaming profiles on your network and you need to track a particular INI setting.

In the INI Settings tree, select the setting you want to track > check Track Distribution Per User.

Distribute Always

Use this option if you want to create or delete a setting every time the application is distributed.

In the INI Settings tree, select the setting to always distribute > check Distribute Always.

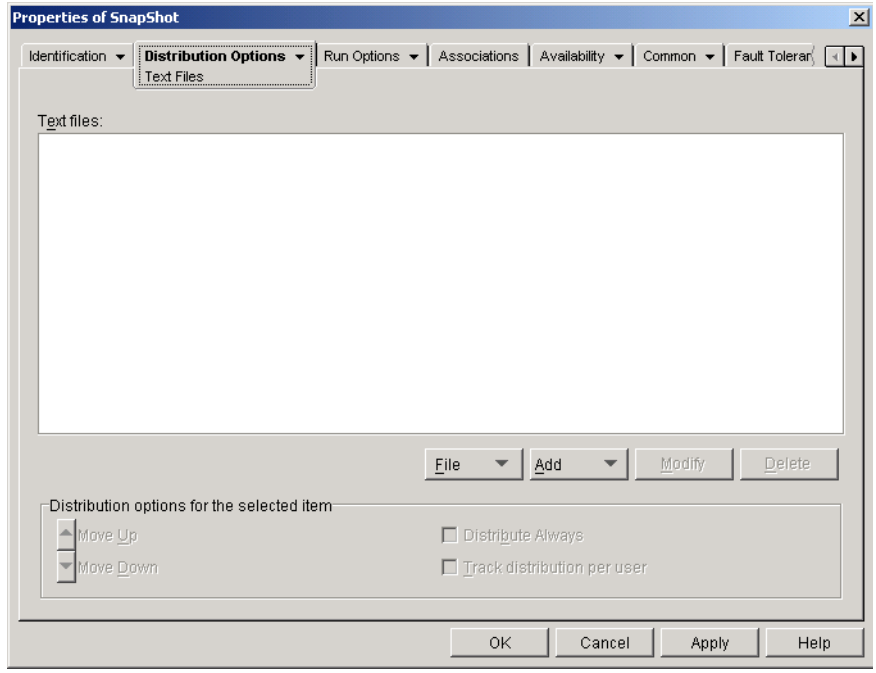
Move Up / Move Down

Use these options to position sections and values in the order in which you want them to be created, modified, or deleted.

In the INI Settings tree, select the section or value to move > click Move Up or Move Down.

Text Files Page (.AOT/.AXT Application Object Only)

The Text Files property page, shown below, determines the modifications that Application Launcher/Explorer will make to text files (such as CONFIG.SYS and AUTOEXEC.BAT) when distributing the application to a workstation.



Text Files

The Text Files tree shows all the text files that will be modified or added when the application is distributed. Each modification to a file is displayed subordinate to the file.

File

Click File > Find to search for specific files or information in the Text Files tree.

Click File > Import to import text files from another Application object's .AOT or .AXT file.

Add

This option lets you add text file modifications to the Text Files tree. Only the modifications displayed in the Text Files tree will be made when the application is distributed.

- **File:** To add a text file to the tree, click Add > File to create the file entry. You can type the filename or the path and filename (for example, AUTOEXEC.BAT or C:\AUTOEXEC.BAT). Only local workstation drives, UNC server paths, and macros are valid.

  In general, you should specify a path if possible. If you enter the filename only, Application Launcher/Explorer will search all directories specified in the workstation's PATH environment variable. If it does not find a matching filename, it will assume the file doesn't exist and create it in the first directory specified in the PATH variable.

- **Change:** To add a change to a file that is in the Text Files list, select the file, click Add > Change to display the Edit Text File dialog box. Make the desired changes. Click Help in the Edit Text File dialog box for information about each of the dialog box fields.

  You can add multiple modifications to a text file. For example, you may want to make one modification that replaces text in the file and another modification that adds text to the end of the file. Each modification you add is displayed beneath the text file in the Text Files list.

Modify

To change the name of a text file, select the file in the Text Files tree > click Modify > enter the new name.

To edit one of the text file's modifications, select the modification in the Text Files tree > click Modify to display the Edit Text File dialog box > make the desired changes. Click Help in the Edit Text File dialog box for information about each of the dialog box fields.

Delete

In the Text File tree, select the text file or text file modification you want to delete > click Delete.

Distribution Options

Use these options to set individual distribution options for text files and text file modifications. The options change depending on whether you have selected a text file or a text file modification in the Text Files tree.

No Reboot Needed If Modified

This option appears only when you have selected a text file. Check this option if you don't want users to reboot after you make changes to the selected text file. The Reboot and Prompt Reboot options on the Distribution Options > Options page override this setting.

Move Up / Move Down

These options appear only when you have selected a text file modification. Click Move Up or Move Down to position the modification according to the order in which you want it applied.

Distribute Always

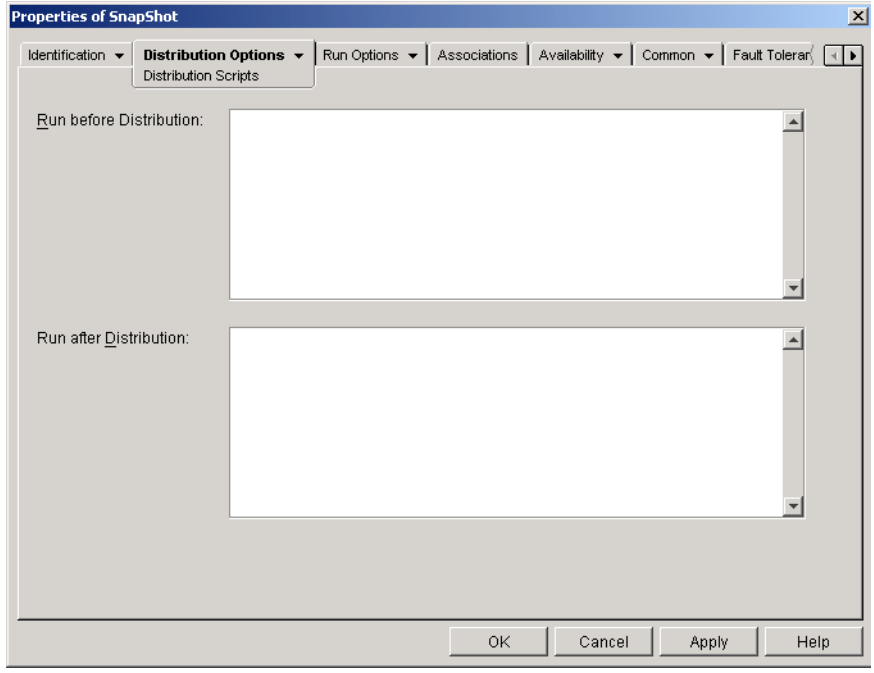
This option appears only when you have selected a text file modification. Check this option if you want the selected modification to be applied every time the application distributes.

Track Distribution Per User

This option appears only when you have selected a text file modification. Check this option if you have enabled users with roaming profiles on your network and you need to track the selected modification.

Distribution Scripts Page

The Distribution Scripts property page, shown below, defines the scripts Application Launcher/Explorer will execute when distributing the application.



The scripts support the same commands and syntax as the Novell Client™, with the exception of the ones listed below. For script commands, syntax, and examples, see the Novell Client documentation (http://www.novell.com/documentation/lg/noclienu/docui/index.html) on the Novell Documentation Web site (http://www.novell.com/documentation).

Application Launcher/Explorer will not output anything to the screen, display errors, or support the following commands:

CLS
DISPLAY
EXIT
FDISPLAY
INCLUDE
LASTLOGINTIME
NO_DEFAULT
NOSWAP
PAUSE
PCOMPATIBLE
SCRIPT_SERVER

SET_TIME
SWAP
WRITE

Run Before Distribution

Use this text window to enter any script commands you want executed before the application is distributed. The script is executed in the order shown below:

1. Run Before Launching script executed (Run Options > Launch Scripts page)
2. Run Before Distribution script executed
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed
5. Application launched
6. Application closed (by user)
7. Run After Termination executed (Run Options > Launch Scripts page)

Run After Distribution

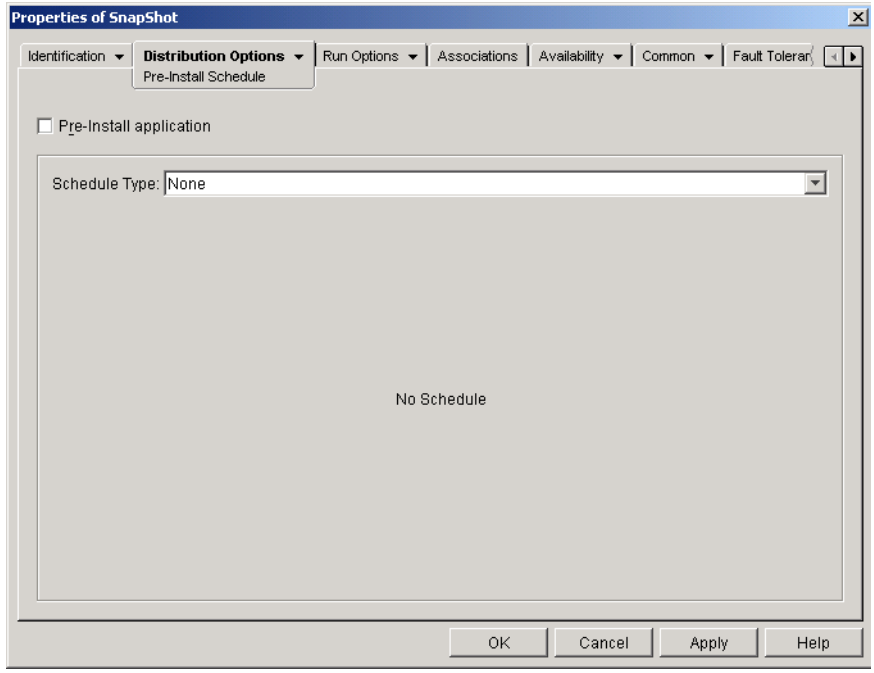
Use this text window to enter any script commands you want executed after the application is distributed. The script is executed in the order shown below:

1. Run Before Launching script executed (Run Options > Launch Scripts page)
2. Run Before Distribution script executed
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed
5. Application launched
6. Application closed (by user)
7. Run After Termination executed (Run Options > Launch Scripts page)

Pre-Install Schedule Page (.AOT/.AXT Application Object Only)

The Pre-Install Schedule property page, shown below, enables Application Launcher/Explorer to install the application to a workstation before the user launches the Application object the first time. Because you can specify when

the installation will take place, this enables an off-line, or “lights-out,” distribution of the application and saves the user the wait associated with installing the application. For example, you could pre-install the application after work hours so the application is ready to use when the user comes to work.



With a pre-install, Application Launcher/Explorer performs all workstation-related distribution processes (file copying, modifying text files, .INI files, and workstation registry settings). Then, when the user double-clicks the Application object, Application Launcher/Explorer finishes the installation by performing any user-specific distribution processes (modifying user registry keys, etc.).

You can pre-install an application that is associated with either workstations or users. For user-associated applications, the user must be logged in and Application Launcher/Explorer must be running. For workstation-associated applications, a user does not need to be logged in.

IMPORTANT: On Windows NT/2000 workstations, if a user is not logged in, the user-specific macros will point to the default user directories. This scenario affects the ability to place folders and icons on the Start menu. There are two ways to solve this issue: 1) Mark the macro entries in the Application object as Track Distribution

Per User or 2) Change the user-specific macro to an All Users macro (i.e. %*PROGRAMS% to %*COMMONPROGRAMS%).

Pre-Install Application

Check this option to enable the application to be pre-installed. If you don't check this option, the application will not be pre-installed, even if you establish a schedule.

Schedule Type

Select the type of schedule you want to use. You can choose None, Specified Days, or Range of Days.

None

Use this option to indicate no schedule. The application will be pre-installed as soon as the system requirements have been established (Availability > System Requirements page) and the application has been associated with the object (Associations page).

Specified Days

Use this option to select specific dates when you want to pre-install the application. You cannot select more than 350 specific dates.

- **Date Range:** The Date Range list displays all dates when the application will be pre-installed. To add a date, click Add > select the date you want > click OK to display it in the list.
- **Times for Selected Dates:** Select the availability start time and end time. The times apply to all dates in the Date Range list.
- **Spread from Start Time (in Minutes):** Enter a number of minutes between available times if you don't want multiple workstations installing the application at once for fear of overloading the network. The spread option spreads out access times over the number of minutes specified so that all pre-installations don't occur at the same time.

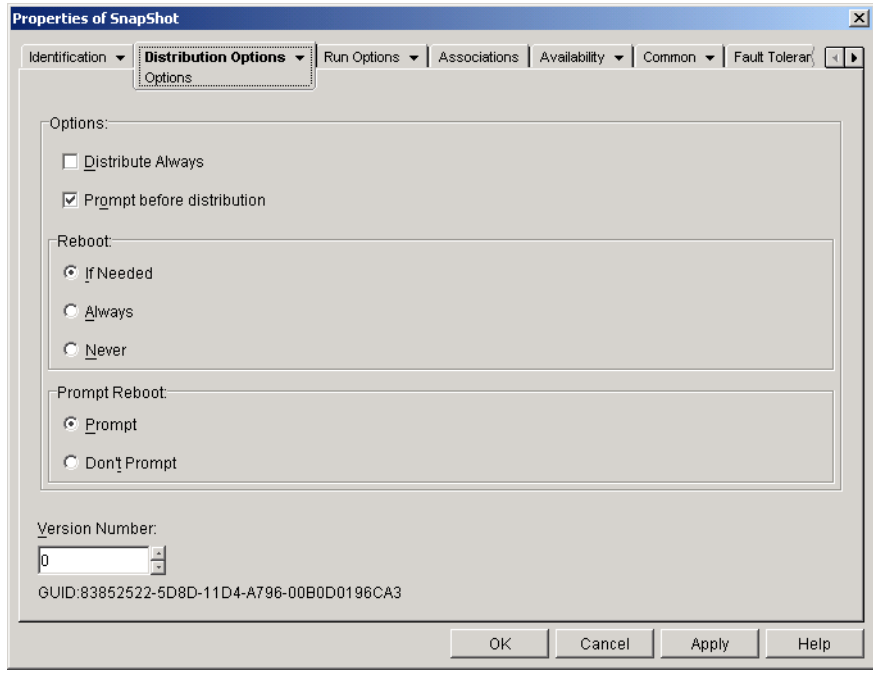
Range of Days

Use this option to select a range of days to pre-install the application. You can also use this option to pre-install the application only on certain days of the week within a given range of dates.

- **Date Range:** To define the range of days, select a start date and an end date > select the days (Sunday through Saturday) within the established date range.
- **Times for Selected Range:** Select the availability start time and end time. The times apply to each day in the established date range.
- **Spread from Start Time (in Minutes):** Enter a number of minutes between available times if you don't want multiple workstations installing the application at once for fear of overloading the network. The spread option spreads out access times over the number of minutes specified so that all pre-installations don't occur at the same time.
- **Use the Schedule in GMT for All Clients:** The schedule is based on the workstation's time zone. If your network spans different time zones and you schedule an application to run at 1:00 p.m., it runs at 1:00 p.m. in each time zone. You can select this option to have workstations run applications at the same time regardless of their time zones (for example, 1:00 p.m. Rome time and 11:00 p.m. Los Angeles time).

Options Page

The Options property page, shown below, determines general options to be used by Application Launcher/Explorer when distributing the application to a workstation.



Options

These options let you determine if the application should always be distributed and if the user should be prompted to accept or reject the distribution.

Distribute Always

Check this option to force a distribution of the entire Application object every time the user runs the application or when the application is set for a force run on the workstation. This option is useful to ensure that all application settings are updated every time the application runs.

If you need only specific Application object settings to be distributed each time, you can update settings on a case-by-case basis. For example, if you want to always distribute a particular registry key and value, you can set the Distribute Always option on the Registry Settings page (Distribution Options tab) for that particular key and value.

This setting overrides the Distribute Always option on the Registry Settings, INI Settings, Application Files, Icons/Shortcuts, and Text Files pages (Distribution Options tab).

Prompt Before Distribution

Check this option to prompt users whether or not they want the application distributed to their workstation. Users are prompted the first time they click the application icon; all subsequent times they are not prompted. To better help users make a decision about installing the application, the prompt includes the text you've entered in the Description page (Identification tab).

Reboot

Select how a workstation reboot should occur. The available options are:

- **If Needed:** Application Launcher/Explorer reboots the workstation if changes need to be made that cannot occur while Windows is running (such as replacing open DLLs).
- **Always:** Application Launcher/Explorer always reboots the workstation after distributing the application.
- **Never:** Application Launcher/Explorer does not reboot the workstation. The changes take effect the next time the workstation reboots.

Prompt Reboot

Select whether or not the user is prompted to reboot the workstation.

Version Number

The version number is a unique number between 0 and 65535 that you can increment as you make revisions to the Application object. If you increment the version number, the application is distributed again the next time the user launches the Application object.

GUID

The GUID (global unique identifier) is generated using the version number and is changed each time you change the version number. When Application Launcher/Explorer distributes an application to a workstation, it adds the GUID to the registry's HKEY_LOCAL_MACHINE for tracking purposes.

You can use the GUID to track and troubleshoot distributed applications. For example, if you want to ensure that a particular application has been distributed to a workstation, you can compare the GUID as recorded in the Application object's Distribution page with the GUID that is currently included in the workstation's registry. Using the Synchronize Distributed

GUIDs option (Tools > Application Launcher Tools menu), you can make several Application objects share the same GUID. This is useful if you are distributing a suite of applications. Using the Generate New GUIDs option (Tools > Application Launcher Tools menu), you can also ensure that two or more applications have different GUIDs.

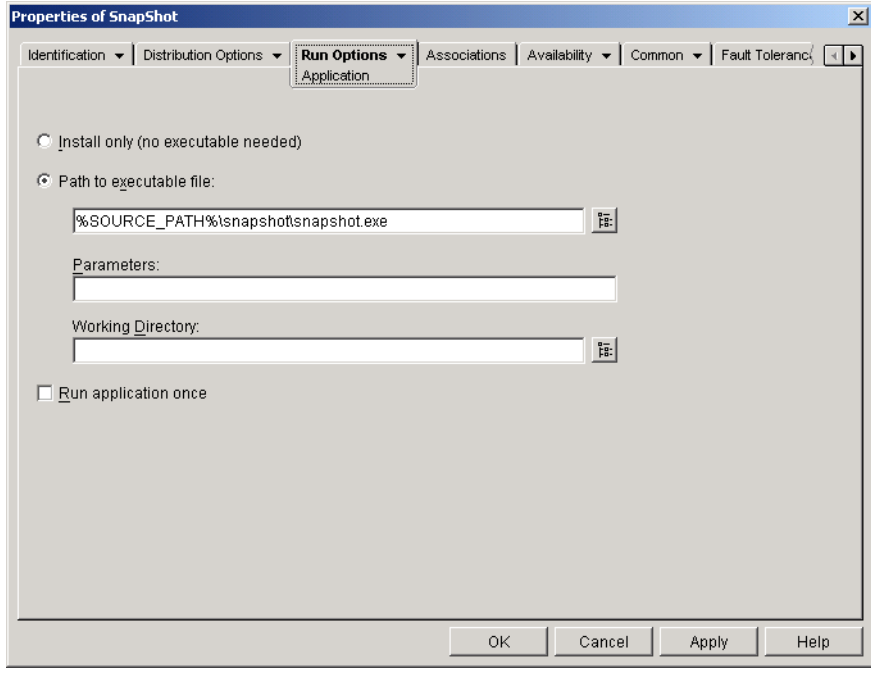
Run Options Tab

The Run Options tab includes the following pages to help you configure how the Application object runs on the workstation:

- "Application Page" on page 155
- "Environment Page" on page 158
- "Launch Scripts Page" on page 160
- "Environment Variables Page" on page 163
- "License/Metering Page" on page 164

Application Page

The Application property page, shown below, determines general options to be used by Application Launcher/Explorer when running the application on a workstation.



Install Only (No Executable Needed)

Check this option if the Application object does not include an application executable that Application Launcher/Explorer should run after distributing the Application object. For example, you would want to use this option if the Application object's only purpose is to update some files on the workstation, such as a driver or font, or to make modifications to some workstation settings, such as the registry settings.

Path to Executable File

Enter, or browse for, the path to the executable that Application Launcher/Explorer will run after the application is distributed. Typically, this path is defined when you create the Application object and should already contain the path to where the executable is located on the network or where it will be located on the user's workstation after distribution.

The following path syntaxes are valid:

*server**volume*:*path*
*server**volume**path*

volume_object_name:*path*
directory_map_object_name:*path*
driveletter:*path*

You can also use macros in this field. For information about macros, see Chapter 13, “Macros,” on page 217.

Parameters

Specify any command line parameters that need to be passed to the executable specified in the Path to Executable File field. See your Windows documentation for more information.

You can also use macros in this field. For information about macros, see Chapter 13, “Macros,” on page 217.

Working Directory

Specify the working directory of the executable you specified in the Path to Executable File field. Typically, this is the directory in which the executable file is located. However, if another directory contains additional files required by the application to run, specify that directory. See your Windows documentation for more information.

The following path syntaxes are valid:

*server**volume*:*path*
*server**volume**path*
volume_object_name:*path*
directory_map_object_name:*path*
driveletter:*path*

You can also use macros in this field. For information about macros, see Chapter 13, “Macros,” on page 217.

Run Application Once

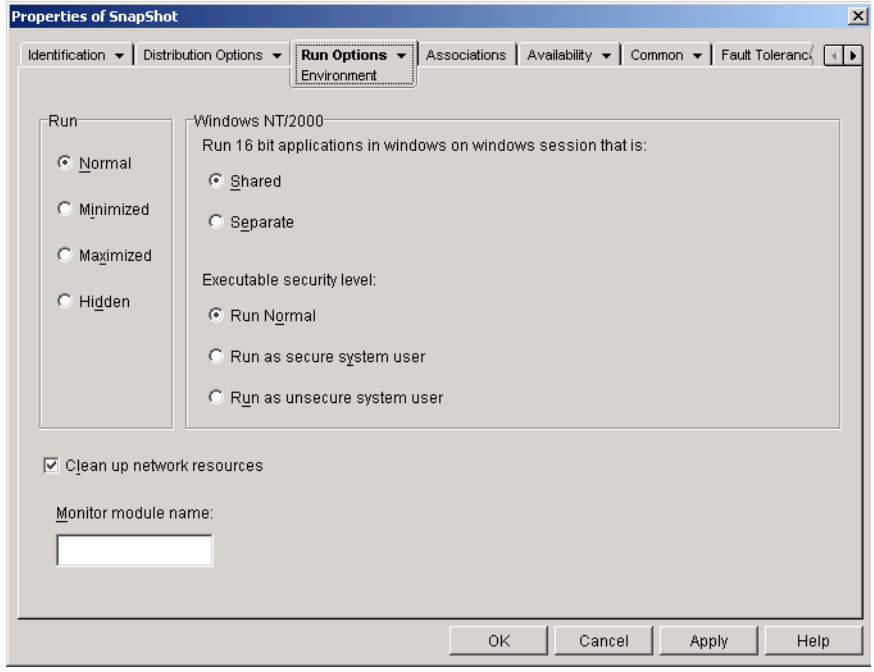
Check this option to have Application Launcher/Explorer run the application one time only and then remove the Application object's icon from the workstation. If the Application object's version number (Distribution Options > Options page) changes, the Application object icon will appear again so that the new version can be distributed.

This option can be useful when an Application object's only purpose is to install software to a workstation. Having the Application object's icon remain after the application has been installed can be confusing to users, especially if you have an application shortcut or program group icon created during the distribution (see the Distribution Options > Icons/Shortcuts page).

This option can also be used in combination with other Application object settings. For example, if the application is specified as Force Run on the Associations page (meaning that without any user involvement, the application will run when it becomes available to Application Launcher/ Explorer), the application is forced to run only one time.

Environment Page

The Environment property page, shown below, specifies the workstation environment that Application Launcher/Explorer will create before running the application on the workstation.



Run

Select the application's initial window size: Normal, Minimized, Maximized, or Hidden. In Hidden mode, the application will run normally without a user interface available. This is useful if you want the application to process something then go away without user intervention. For example, you could use it for the Remote Control agent if you don't want it to appear in the taskbar.

Windows NT/2000

Use these options to specify how applications run on Windows NT/2000 workstations.

Run 16-Bit Applications in Windows on Windows Session

If you are setting up a 16-bit application to run on Windows NT/2000, specify either a shared or separate Windows on Windows (WOW) session. To run on Windows NT/2000, 16-bit applications must run in a Virtual DOS Machine (VDM) that emulates Windows 3.1 functionality. 16-bit Windows applications can run in their own separate WOW sessions or they can share a WOW session with other 16-bit applications. Sharing a WOW session saves memory and allows applications to use DDE or OLE to communicate with other applications in the same WOW session. Running an application in its own separate WOW session allows for more robust operation because the application is isolated from other 16-bit applications. Because some applications do not run properly in a separate WOW session, the default option is to run the 16-bit Windows application in a shared WOW session.

Executable Security Level

Select one of the following security levels for the application:

- **Run Normal:** The application inherits the logged-in user's credentials. For example, the application has the same rights to the registry and the file system as the logged-in user.
- **Run As Secure System User:** The application inherits the workstation's credentials and is run as a Windows NT/2000 process. For example, the application has full rights to the registry and the file system. No user interface is available and the application is only visible in the Task Manager. This option is useful when distributing applications that require full access to the workstation but require no user intervention, such as a service pack or update.

- **Run as Unsecure System User:** This option is the same as Run As Secure System User, with the exception that the user interface is available. This means that the application can display dialog boxes and prompt the user for information if necessary. In return, the user will have the ability to do whatever that application offers on the workstation. For example, if you specify REGEDIT as an unsecure system user, the workstation's user will able to edit any registry values. The unsecure system user does not currently work on Citrix* or Windows NT Terminal Server.

Clean Up Network Resources

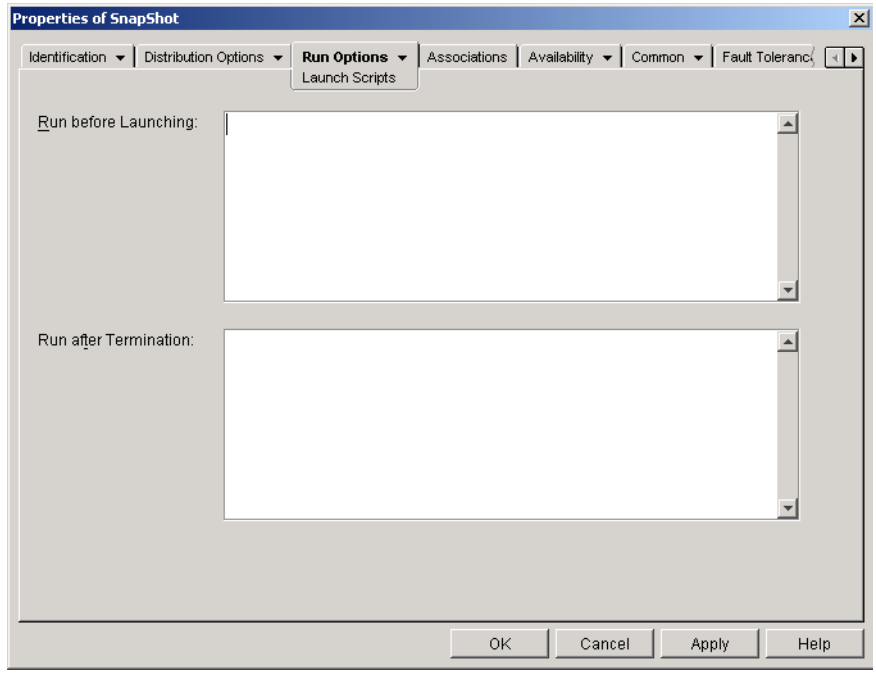
Check this option to remove network connections, drive mappings, and port captures that were established for running the application. If this option is not selected, the network resources remain in effect. If the resource is already in use when Application Launcher/Explorer starts the application, it is not cleaned up until all other distributed applications are finished using it. If distributed applications are still running when Application Launcher/Explorer is terminated, the allocated resources remain intact.

Monitor Module Name

When an application is launched, Application Launcher/Explorer monitors the executable of the application. When the executable terminates, the process of cleaning up network resources begins. However, it's possible that the executable filename is actually a "wrapper" that sets up environments, runs other executables, and then terminates. If Application Launcher/Explorer monitors the wrapper executable, it might prematurely start cleaning up network resources before the application has terminated. Consult your application documentation about whether the application uses a wrapper executable. If it does, find out the name of the module that remains running. Type this name (without the extension) in the text box provided.

Launch Scripts Page

The Launch Scripts property page, shown below, defines the scripts Application Launcher/Explorer will execute when running the application.



The scripts support the same commands and syntax as the Novell Client, with the exception of the ones listed below. For script commands, syntax, and examples, see the Novell Client documentation (http://www.novell.com/documentation/lg/noclienu/docui/index.html) on the Novell Documentation Web site (http://www.novell.com/documentation).

Application Launcher/Explorer will not output anything to the screen, display errors, or support the following commands:

CLS
DISPLAY
EXIT
FDISPLAY
INCLUDE
LASTLOGINTIME
NO_DEFAULT
NOSWAP
PAUSE
PCOMPATIBLE
SCRIPT_SERVER

SET_TIME
SWAP
WRITE

Run Before Launching

Use this text window to enter any script commands you want Application Launcher/Explorer to execute before launching the application. The script is executed in the order shown below:

1. Run Before Launching script executed
2. Run Before Distribution script executed (Distribution Options > Distribution Scripts page)
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed (Distribution Options > Distribution Scripts page)
5. Application launched
6. Application closed (by user)
7. Run After Termination executed

Run After Termination

Use this text window to enter any script commands you want Application Launcher/Explorer to execute after the user closes the application. The script is executed in the order shown below:

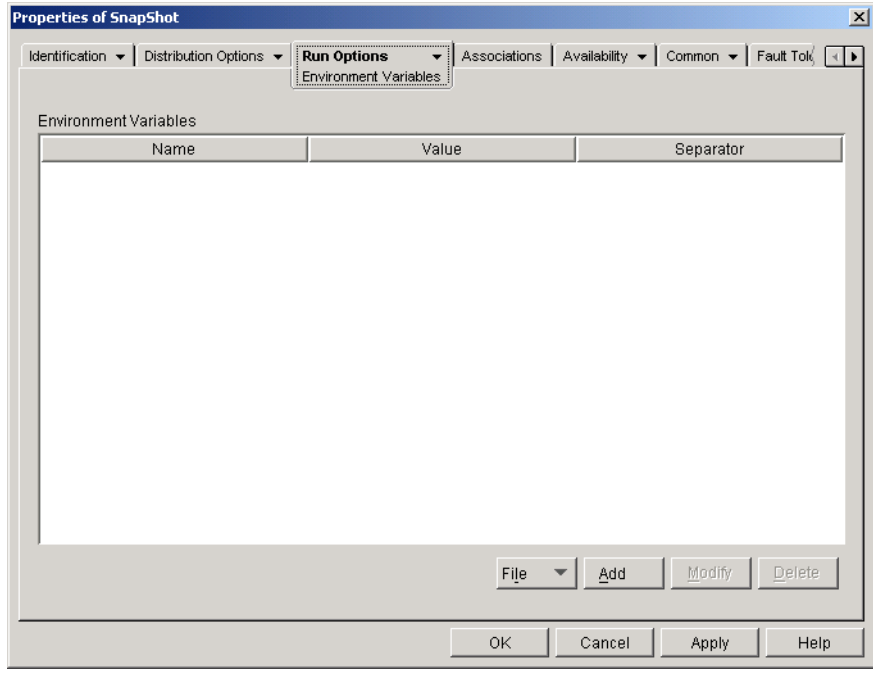
1. Run Before Launching script executed
2. Run Before Distribution script executed (Distribution Options > Distribution Scripts page)
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed (Distribution Options > Distribution Scripts page)
5. Application launched
6. Application closed (by user)
7. Run After Termination script executed

IMPORTANT: The Run After Termination script is executed before the network resources are cleaned up (Run Options > Environment page). However, the Clean

Up Network Resources option will not clean up drive mappings or port captures created by the Run Before Launching script. To clean up these resources, enter the appropriate commands in the Run After Termination script.

Environment Variables Page

The Environment Variables property page, shown below, specifies the environment variables that Application Launcher/Explorer will set before running the application. For example, you can specify the application's working directory in the PATH environment variable.



Environment Variables

The Environment Variables list displays all variables that will be modified before the application is run. Application Launcher/Explorer modifies only the variables in this list.

File

Click File > Find to search for variables or variable data in the Environment Variables list.

Click File > Import to import variables from another Application object's .AOT or .AXT file.

Add

To add a variable to the Environment Variables list, click Add to display the Edit Environment Variables dialog box > enter the variable name and data > check Append to Existing Data Using Separator Character (only if you want to append the data to the workstation's existing variable) and enter a separator character > click OK.

Modify

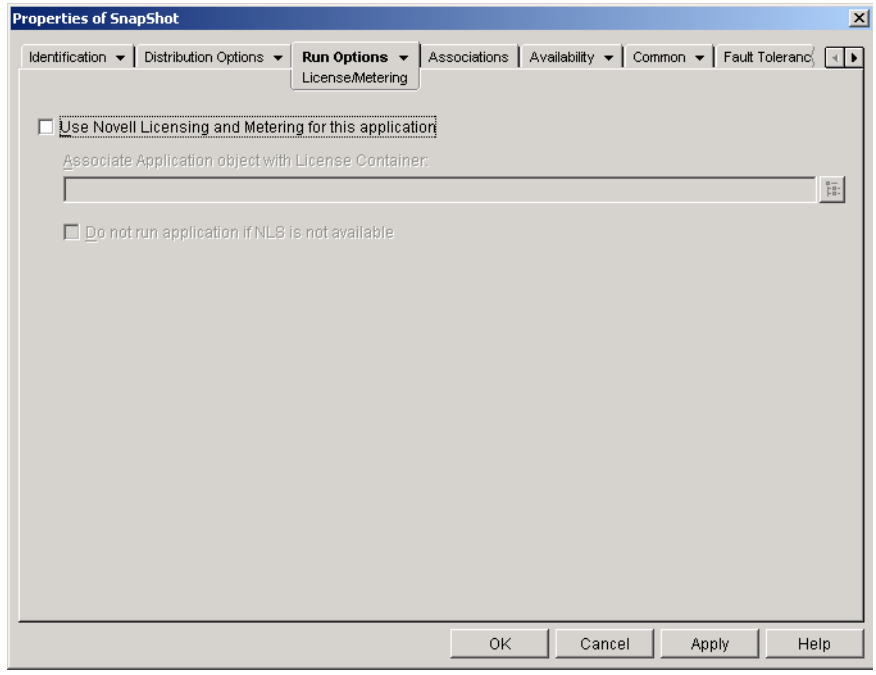
In the Environment Variables list, select the variable you want to modify > click Modify.

Delete

In the Environment Variables list, select the variable you want to delete > click Delete to remove it from the list.

License/Metering Page

The License/Metering property page, shown below, configures Application Launcher/Explorer to use Novell Licensing Services (NLS) to track application usage and comply with the application's licensing agreement. NLS must already be installed, and you must have created a License Container and a Metered Certificate for the application. For detailed information about setting up license metering, see Chapter 10, "Metering Software Licenses," on page 115.



Use Novell Licensing and Metering for this Application

Check this option to enable metering of the application.

Associate Application Object with License Container

Once metering is enabled, select the License Container in which you've created the application's Metered Certificate.

Do Not Run Application If NLS Is Not Available

Check this option to prevent users from running the application if NLS is not running.

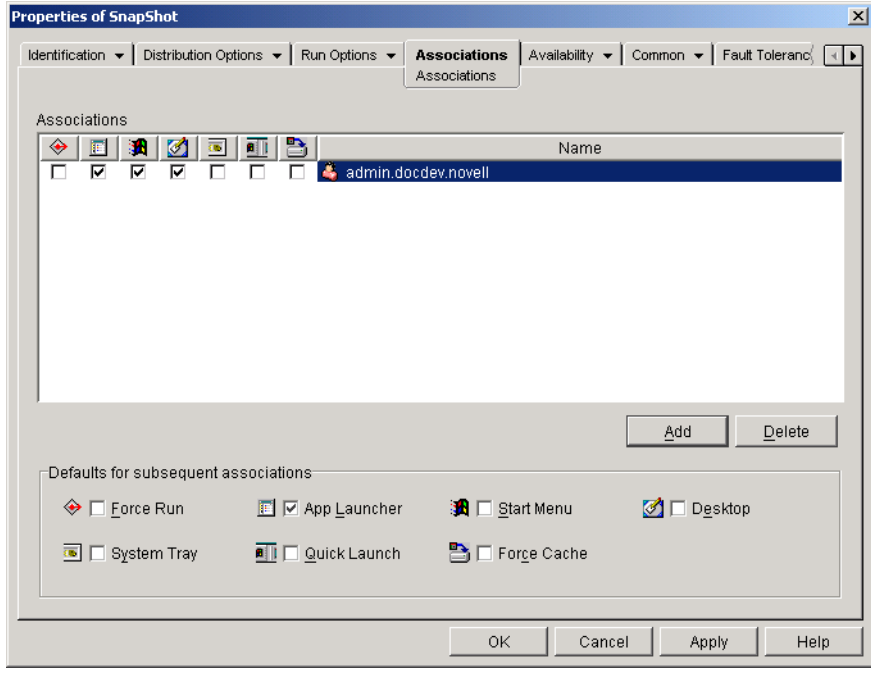
Associations Tab

The Associations tab includes one page only. This page lets you associate the Application object with users and workstations.

- "Associations Page" on page 166

Associations Page

The Associations property page, shown below, lists the objects that have been associated with the Application object. Objects that are associated with the Application object can see and use the application. You can grant rights to User, Workstation, Group, Workstation Group, Organizational Unit, Organization, and Country objects.



Associations

The Associations list displays the objects associated with the Application object and lets you add or delete objects.

Add

To associate an object with the Application object, click Add > browse and select the object > click OK. All objects receive the default characteristics (Force Run, App Launcher, Start Menu, Desktop, System Tray, Quick Launch, Force Cache) selected in the Defaults for Subsequent Associations list. After you add an object to the Associations list, you can modify its characteristics by checking or unchecking the appropriate boxes.

Delete

To disassociate an object from the Application object, select the object in the Associations list > click Delete.

Defaults for Subsequent Associations

When you add an object to the Associations list, the object receives the default characteristics selected in this box. If you change the characteristics that are selected, any objects you add after that will have the new characteristics.

Force Run

Runs the Application object as soon as Application Launcher/Explorer or Workstation Helper starts and the Application object is available.

You can use the Force Run option in conjunction with several other settings to achieve unique behaviors. For example, if you use the Force Run option with the Run Application Once option (Run Options tab > Applications page), as soon as the Application object is distributed it will be run one time and then removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a pre-determined time. If so, select Force Run, select the Run Application Once option on the Application page (Run Options tab), and define a schedule using the Schedule page (Availability tab).

If you want to force run several applications in a specific order, mark each of them as Force Run. Then give each Application object a numeric order by using the Order Icon Display option on the Icon page (Identification tab).

IMPORTANT: When associating an Application object with a workstation, Force Run will not work if the Application object uses a Microsoft Windows Installer (.MSI) package. This is true only when the Application object is associated with a workstation. Force Run works with .MSI Application objects associated with users.

App Launcher

Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu

If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be added to the top of the Start menu, unless you assign the Application object to

a folder and use the folder structure on the Start menu. See the Application object's Folders page (Identification tab).

Desktop

If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows desktop.

System Tray

If the workstation is running Application Explorer, this option displays the Application object's icon in the Windows system tray.

Quick Launch

Displays the Application object's icon on the Windows Quick Launch toolbar.

Force Cache

Forces the application source files and other files required for installation to be copied to the workstation's cache directory. The user can then install or repair the application while disconnected from NDS. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see Chapter 7, "Running in Disconnected Mode," on page 75.

IMPORTANT: This option is required only if you want to ensure that the user can install or repair the application while disconnected from NDS. Without this option selected, the user will still be able to launch the application in disconnected mode, provided the application has already been distributed (installed) to the workstation.

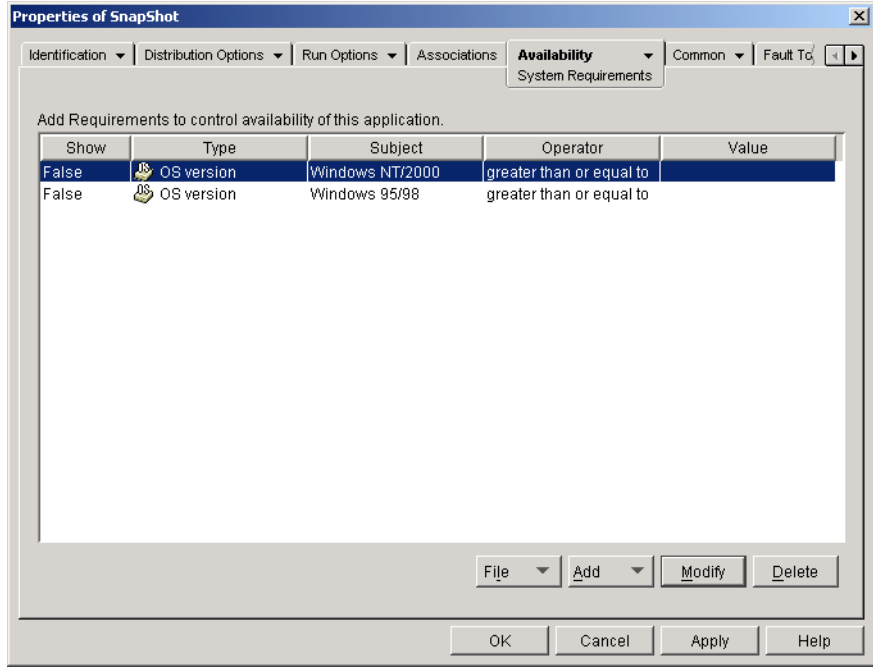
Availability Tab

The Availability tab includes the following pages to help you determine the system requirements a workstation must meet before the application will be distributed to the workstation, the days and times the application will be available to users, and the messages and warnings Application Launcher/ Explorer will display to users when terminating an application that has become unavailable to users:

- "System Requirements Page" on page 169
- "Schedule Page" on page 184
- "Termination Page" on page 185

System Requirements Page

The System Requirements property page, shown below, determines the system requirements a workstation must meet before Application Launcher/ Explorer will distribute the application to it.



Add Requirements to Control Availability of This Application

This list displays all the requirements associated with the distribution of the application. For each requirement, the following information is displayed:

- **Show:** This column indicates if the Application object's icon will be displayed on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met.

  If an icon is displayed in a disabled state, users can right-click the icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

- **Type:** This column shows the category of the system requirement. Categories are explained under Add below.
- **Subject:** This column shows the subject of the system requirement. For example, if the type is OS Version, the subject could be Windows 95/98 or Windows NT/2000.
- **Operator:** This column shows if a condition (greater than, less than, equal to, etc.) must be met for the assigned value.
- **Value:** This column shows if a value is associated with the subject. For example, if Windows NT/2000 is the subject, the version may need to be greater than or equal to 4.00.950 in order for the application to be distributed.

File

Click File > Find to search for requirements that include certain information. For example, you could search for requirements that include "Windows," "greater than," or "system memory."

Click File > Import to import requirements from another Application object.

Add

Click this option to add a new system requirement, based on one of the following categories:

- **Applications:** Require a specific application to be installed or not be installed. The application must have been distributed through an Application object. For more information, see “Applications” on page 171.
- **Disk Space:** Require a specific amount of disk space on the workstation. For more information, see “Disk Space” on page 172.
- **Environment Variables:** Require specific environment variables to be present or not present. For more information, see “Environment Variables” on page 173.
- **Memory:** Require a specific amount of memory on the workstation. For more information, see “Memory” on page 175.
- **Operating System:** Require a specific operating system. For more information, see “Operating System” on page 176.
- **Processor:** Require a specific processor type. For more information, see “Processor” on page 177.

- **Registry:** Require specific registry values to be present or not present. For more information, see “Registry” on page 178.
- **File Existence:** Require a file to be present or not present. For more information, see “File Existence” on page 179.
- **File Version:** Require a file to be a specific version. For more information, see “File Version” on page 180.
- **File Date:** Require a file to be a specific date. For more information, see “File Date” on page 181.
- **File Size:** Require a file to be a specific size. For more information, see “File Size” on page 182.

Modify

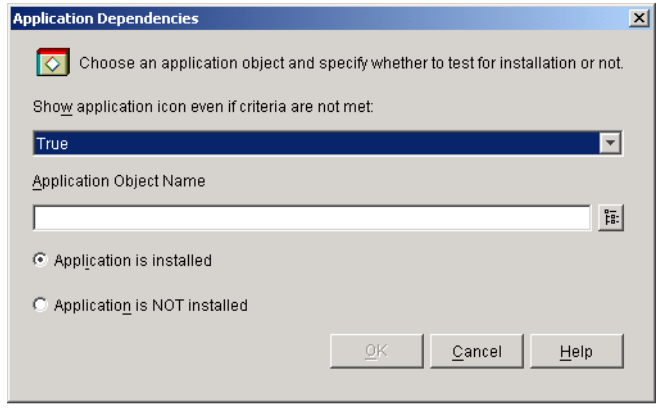
Select a requirement in the list > click Modify to edit the information.

Delete

Select a requirement in the list > click Delete to remove the requirement.

Applications

The Application Dependencies dialog box (System Requirements page > Add > Application) lets you create a requirement based on the existence, or nonexistence, of another application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Application Object Name

Browse and select the appropriate Application object.

Application Is Installed

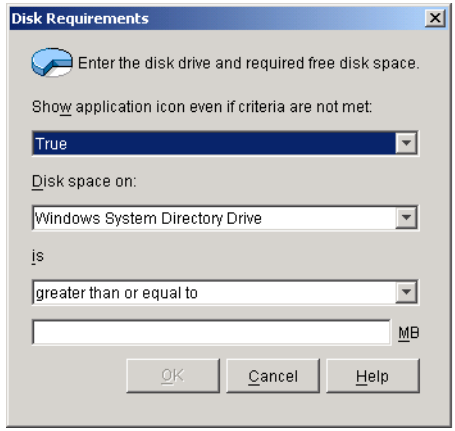
Select this option to indicate that the selected application must already be installed on the workstation.

Application Is Not Installed

Select this option to indicate that the selected application must not be installed on the workstation.

Disk Space

The Disk Requirements dialog box (System Requirements page > Add > Disk Space) lets you create a requirement based on the amount of disk space required by the application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Disk Space On

Select the disk where you are requiring free space. The list includes the following options:

- **A through Z:** Allows you to select the disk that is mapped to drive A, B, C, and so on.
- **Windows System Directory Drive:** Allows you to select the disk where the Windows system directory (for example, WINDOWS\SYSTEM) resides.
- **Windows Directory Drive:** Allows you to select the disk where the Windows directory (for example, WINDOWS) resides.
- **Windows Temp Directory Drive:** Allows you to select the disk where the Windows temporary directory (for example, WINDOWS\TEMP) resides.

If you create multiple disk space requirements for the application, Application Launcher/Explorer adds the requirements together to determine the total disk space requirement. For example, if you specify 20 MB free on the Windows directory drive (which happens to be the C: drive), 10 MB free on the TEMP drive (which is also the C: drive), and 50 MB free on the D: drive, Application Launcher/Explorer distributes the Application object only if there is 30 MB available on the C: drive and 50 MB free on the D: drive.

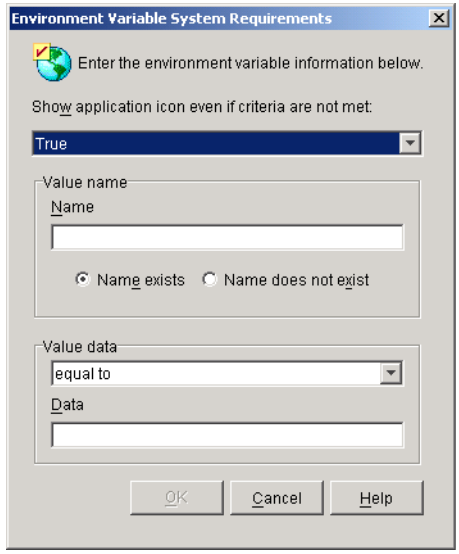
Is (Conditional Statement)

Select whether the available disk space must be less than, less than or equal to, equal to, greater than, greater than or equal to the amount listed in the MB field. For example, if you want to require at least 50 MB of free disk space, select Greater Than or Equal To and then enter 50 in the MB field.

MB (Condition)

Enter the disk space requirement.

The Environment Variable System Requirements dialog box (System Requirements page > Add > Environment Variables) lets you create a requirement based on the existence, or nonexistence, of a specific environment variable. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Value Name

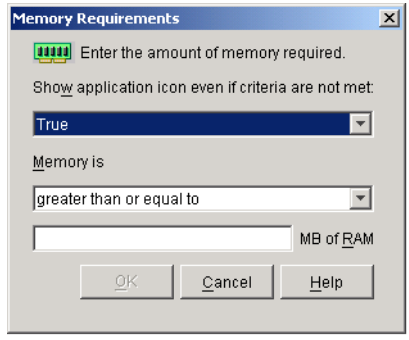
Enter the name of the environment variable. Select the Name Exists option to require the variable to already exist on the workstation, or select the Name Does Not Exist Select option to require the variable to not exist on the workstation.

Value Data

Specify whether the variable data must be equal to, not equal to, contain, or not contain the data you enter in the Data field.

Memory

The Memory Requirements dialog box (System Requirements page > Add > Memory) lets you create a requirement based on the amount of memory (RAM) required to run the application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Memory Is (Conditional Statement)

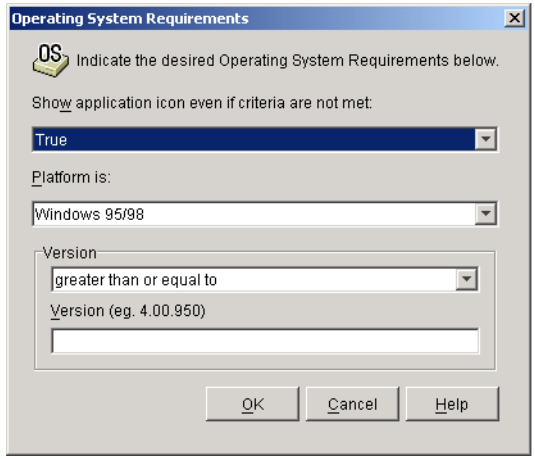
Select whether the total memory must be less than, less than or equal to, equal to, greater than, greater than or equal to the memory entered in the MB of RAM field. For example, if you want to require at least 32 MB of free RAM, select Greater Than or Equal To and then enter 32 in the MB of RAM field.

MB of RAM (Condition)

Enter the required memory amount.

Operating System

The Operating System Requirements dialog box (System Requirements page > Add > Operating System) lets you create a requirement based on the operating system required to run the application. Application Launcher/ Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disable application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

IMPORTANT: With this option set to True, the application icon will be shown only if the operating system platform is the same as the one selected but the version does not meet the version requirement. For example, if you select Windows95/98 as the platform and enter greater than or equal to 4.02.1998 as the version, the icon will not display on a Windows NT/2000 workstation, will display in an active state on a Windows 98 workstation running version 4.02.1998 or newer, and will display in a disabled state on a Windows 95 workstation or a Windows 98 workstation running version 4.01.1998 or older.

Platform Is

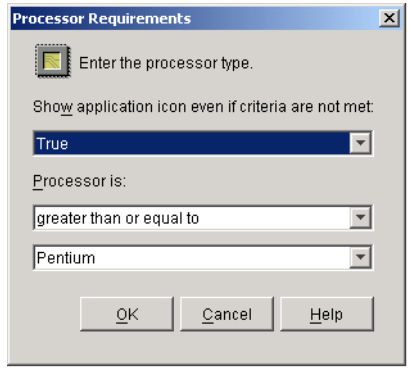
Select the operating system (Windows95/98 or Windows NT/2000) the workstation must be running.

Version

Select whether the version number must be less than, less than or equal to, equal to, greater than, greater than or equal to the number entered in the Version field. For example, if you want to require at least version 4.00.95, select Greater Than or Equal To and then enter 4.00.95 in the Version field.

Processor

The Processor Requirements dialog box lets you create a requirement based on the processor required to run the application. Application Launcher/ Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

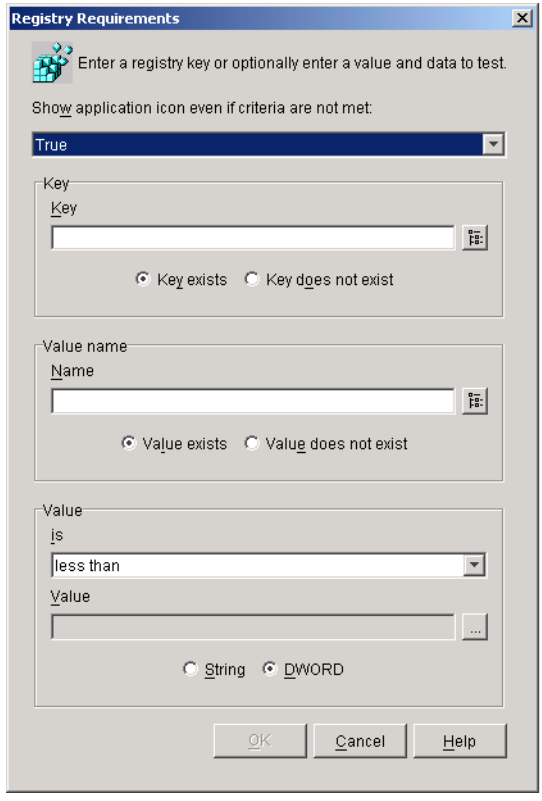
Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Processor Is

Select whether the processor must be less than, less than or equal to, equal to, greater than, or greater than or equal to the processor type (386, 486, Pentium*, Pentium Pro, Pentium II, Pentium III) selected in the processor list. For example, if you want to require at least a Pentium II processor, select Greater Than or Equal To and then select Pentium II.

Registry

The Registry Requirements dialog box lets you create a requirement based on a specific registry key or value. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state

if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Key

Browse and select the registry key you want to check before distribution. Select Key Exists if the key must exist for the distribution to take place, or select Key Does Not Exist if the key cannot exist for the distribution to take place.

Name

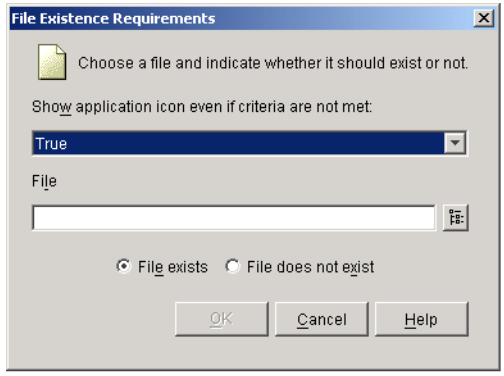
If the key has a specific value you want to check for, browse and select the value. Select Value Exists if the value must exist for the distribution to take place, or select Value Does Not Exist if the value cannot exist for the distribution to take place.

Value

If you specified a value in the Name field, select a condition (less than, less than or equal to, equal to, not equal to, greater than or equal to, greater than) for the value's data > enter the data in the Value field > select whether the value data is a string or DWORD type.

File Existence

The File Existence Requirements dialog box lets you create a requirement based on the existence, or nonexistence, of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file you want to search for.

File Exists

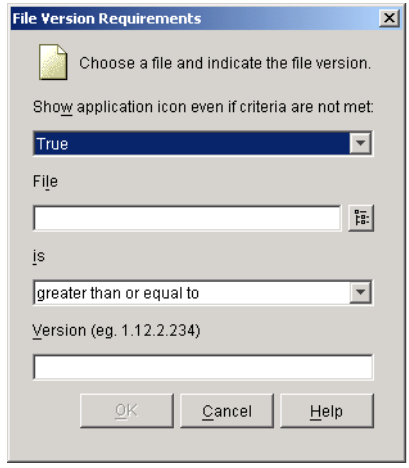
Select this option to require the file to exist on the workstation.

File Does Not Exist

Select this option to require the file to not exist on the workstation.

File Version

The File Version Requirements dialog box lets you create a requirement based on the version of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file whose version you want to check.

Is (Conditional Statement)

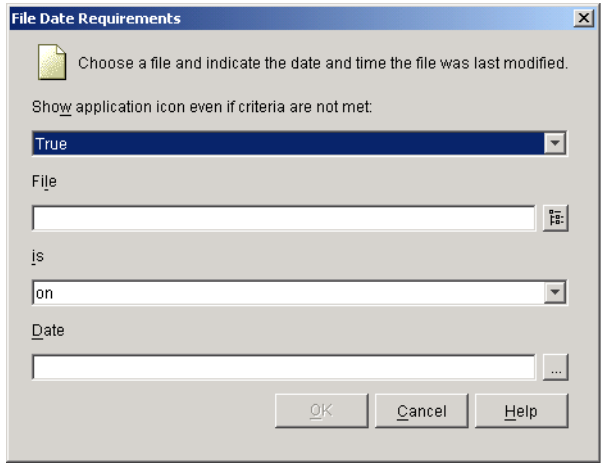
Select whether the version must be less than, less than or equal to, equal to, greater than, greater than or equal to the version entered in the Version field. For example, if you want to require at least version 1.12.2.234, select Greater Than or Equal To and then enter 1.12.2.234 in the Version field.

Version

Enter the file version.

File Date

The File Date Requirements dialog box lets you create a requirement based on the date of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file whose date you want to check.

Is (Conditional Statement)

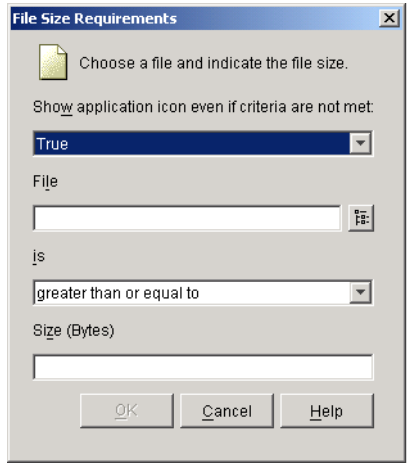
Select whether the file date must be before, on or before, on, on or after, or after the date entered in the Date field. For example, if you want to make sure the application is distributed only if the file is dated 15 January 2000 or older, you would select On or Before and then select the date.

Date

Use the Calendar to select the appropriate date.

File Size

The File Size Requirements dialog box lets you create a requirement based on the size of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file whose size you want to check.

Is (Conditional Statement)

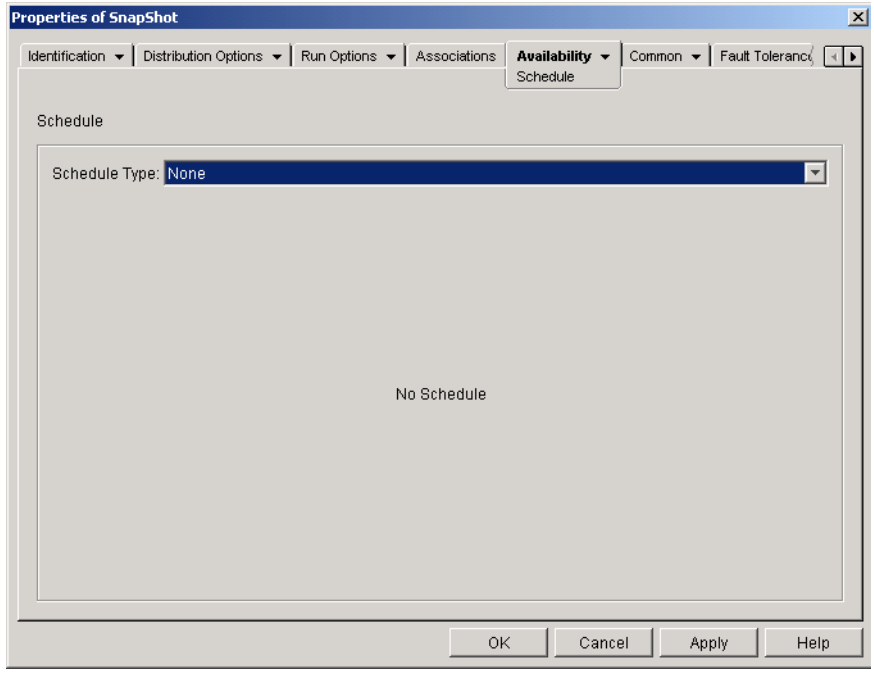
Select whether the size must be less than, less than or equal to, equal to, greater than, or greater than or equal to the size entered in the Size field. For example, if you want to require the file to be at least 200 KB, select Greater Than or Equal To and then enter 200 in the Size field.

Size

Enter the file size in kilobytes.

Schedule Page

The Schedule property page, shown below, defines the dates and times when Application Launcher/Explorer will make the Application object available to users.



Schedule Type

Select the type of schedule you want to use. You can choose None, Specified Days, or Range of Days.

None

Use this option to indicate no schedule. The Application object becomes available to an object as soon as the application's system requirements have been established (Availability > System Requirements page) and the application has been associated with the object (Associations page).

Specified Days

Use this option to select specific dates when you want the Application object to be available. You cannot select more than 350 specific dates.

Date Range: The Date Range list displays all dates when the Application object will be available. To add a date, click Add > select the date you want > click OK to display it in the list.

Times for Selected Dates: Select the availability start time and end time. The times apply to all dates in the Date Range list.

Spread from Start Time (in Minutes): Enter a number of minutes between available times if you don't want all users to run the application at the same time for fear of overloading the network. The spread option spreads out user access times over the number of minutes specified so they don't all run the application at once.

Range of Days

Use this option to select a range of dates to make the application available. You can also use this option to make applications available only on certain days of the week within a given range of dates.

Date Range: To define the range of days, select a start date and an end date, then select the days (Sunday through Saturday) within the established date range that the application will be available.

Times for Selected Range: Select the availability start time and end time. The times apply to each day in the established date range.

Spread from Start Time (in Minutes): Enter a number of minutes between available times if you don't want all users to run the application at the same time for fear of bringing down the network because of the load and traffic. The spread option spreads out user access times over the number of minutes specified so they don't all run the application at once.

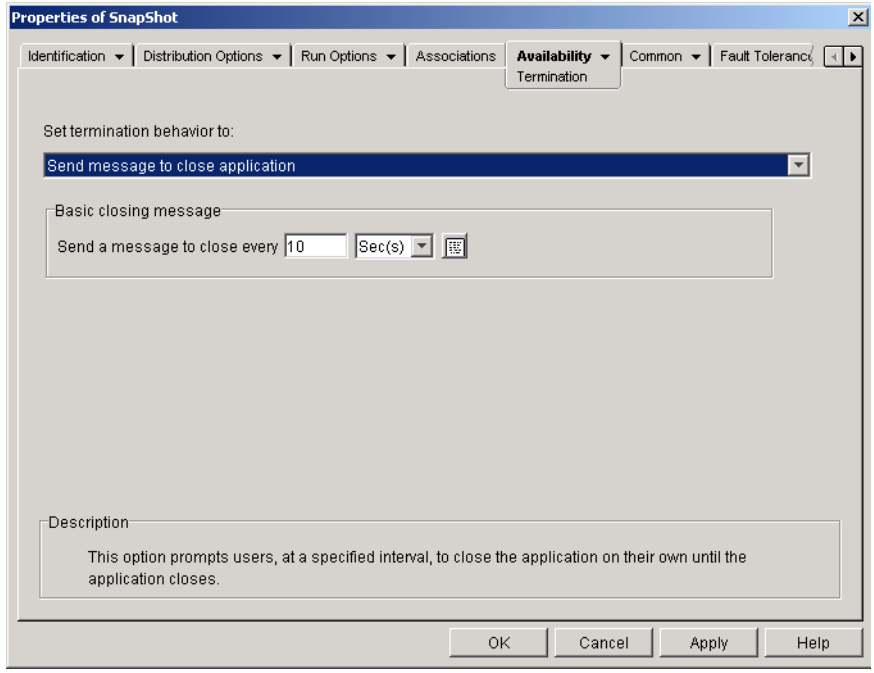
Use the Schedule in GMT for All Clients: The schedule is based on the workstation's time zone. If your network spans different time zones and you schedule an application to run at 1:00 p.m., it runs at 1:00 p.m. in each time zone. You can select this option to have workstations run applications at the same time regardless of their time zones (for example, 1:00 p.m. Rome time and 11:00 p.m. Los Angeles time.

Termination Page

The Termination property page, shown below, determines how Application Launcher/Explorer terminates the application if it becomes unavailable to a

user while the user is running the application. An application becomes unavailable to a user when:

- You disassociate it from the user.
- The availability schedule expires.
- The application's system requirements change and the user's workstation no longer complies.



Set Termination Behavior To

Select the behavior you want used when the application terminates.

None

Select this option to disable any termination behaviors.

Send Message to Close Application

This option causes Application Launcher/Explorer to display a message instructing the user to close the application. The message appears periodically until the user closes the application.

In the Basic Closing Message box, select how often you want Application Launcher/Explorer to display the message. The following default message will be used unless you click the Message button and define a custom message:

```
WARNING! Your access to this application has expired. Please
close the application to avoid losing your work.
```

Send Message to Close Then Prompt to Save Data

This option causes Application Launcher/Explorer to first display a message instructing the user to close the application. If the user does not close the application, Application Launcher/Explorer sends a call to the application instructing it to close. If the user has no unsaved data, the application closes immediately. If the user has unsaved data, the application displays a Save dialog box to allow the user to save the data. Once the application closes, users cannot launch it again.

IMPORTANT: Some applications display a Save dialog box that includes a Cancel button in addition to Yes and No buttons. The Cancel button aborts the close operation and the application remains open. If the application's Save dialog box includes a Cancel button and you want to ensure that the application is closed, you must use the Send Message to Close, Prompt to Save, Then Force to Close option.

Warning Message (Optional): Check the Send Warning option if you want to warn the user to close the application, then enter the number of times to display the warning message and the interval between messages. The following default message will be used unless you click the Message button and define a custom message:

```
WARNING! Your access to this application has expired. Please
close the application to avoid losing your work.
```

Prompt Closing Message: This message is displayed if the user does not close the application after receiving the warning message the specified number of times, or if the warning message option is not enabled.

In the Prompt Closing Message box, enter how often you want to prompt the user to close the application. If, after receiving the prompt, the user does not close the application, Application Launcher/Explorer attempts to close the application. The following default message will be used unless you click the Message button and define a custom message:

```
WARNING! Your access to this application has expired and the
application will close shortly. Please save your work and exit
the application.
```

Send Message to Close, Prompt to Save, Then Force to Close

This option causes Application Launcher/Explorer to first display a message instructing the user to close the application. If the user does not close the application, Application Launcher/Explorer sends a call to the application instructing it to close. If the user has no unsaved data, the application closes immediately. If the user has unsaved data, the application displays a Save dialog box to allow the user to save the data. Regardless of the option (Yes, No, Cancel) the user selects, the application is closed.

Warning Message (Optional): Check the Send Warning option if you want to warn the user to close the application, then enter the number of times to display the warning and the interval between warnings. The following default message will be used unless you click the Message button and define a custom message:

```
WARNING! Your access to this application has expired. Please
close the application to avoid losing your work.
```

Prompt Closing Message: This message is displayed if the user does not close the application after receiving the warning message the specified number of times, or if the warning message option is not enabled.

In the Prompt Closing Message box, enter how often you want to prompt the user to close the application. If, after receiving the prompt, the user does not close the application, Application Launcher/Explorer attempts to close the application. The following default message will be used unless you click the Message button and define a custom message:

```
WARNING! Your access to this application has expired and the
application will close shortly. Please save your work and exit
the application.
```

Application Termination: This message is displayed after Application Launcher/Explorer closes the application. If you don't want to use the default termination message, click the Message button and define a custom message. The default message is:

```
The application you were running has expired. For additional
usage, please contact your system administrator.
```

Send Message to Close Then Force Close with Explanation

This option prompts the user to close the application. If the user does not, Application Launcher/Explorer closes the application automatically and displays a termination message.

Warning Message (Optional): Check the Send Warning option if you want to warn the user to close the application, then enter the number of times to display the warning and the interval between warnings. If the user does not close the application, Application Launcher/Explorer closes the application automatically after the last warning message. The following default message will be used unless you click the Message button and define a custom message:

```
WARNING! Your access to this application has expired. Please
close the application to avoid losing your work.
```

Application Termination: This message is displayed after Application Launcher/Explorer closes the application. If you don't want to use the default termination message, click the Message button and define a custom message. The default message is:

```
The application you were running has expired. For additional
usage, please contact your system administrator.
```

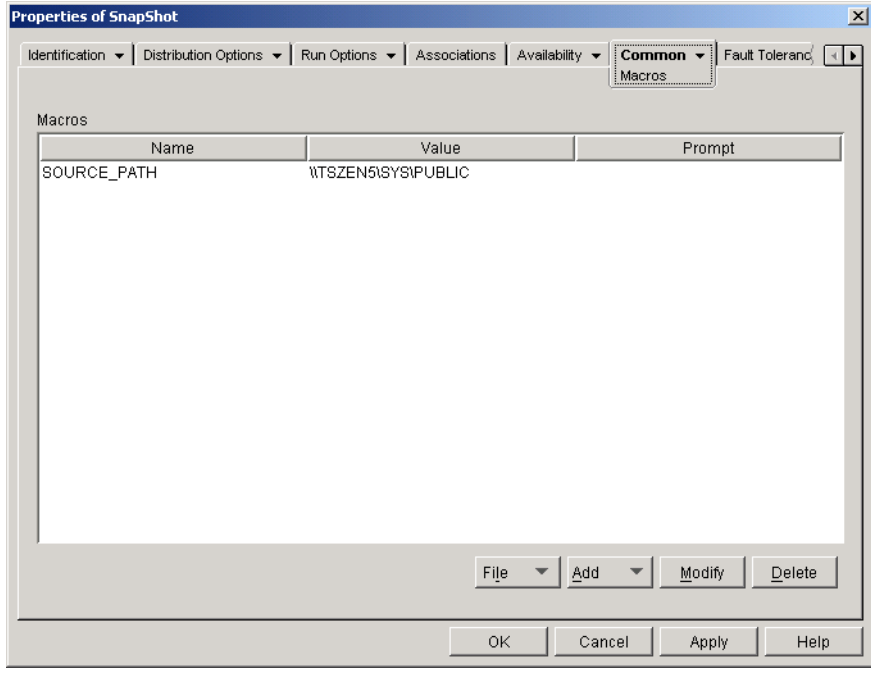
Common Tab

The Common tab includes the following pages:

- "Macros Page" on page 189
- "Drives/Ports Page" on page 193
- "File Rights Page" on page 196
- "Reporting Page" on page 198
- "Properties Page (.MSI Application Object Only)" on page 199
- "Transforms Page (.MSI Application Object Only)" on page 200
- "Imaging Page" on page 202
- "Sources Page" on page 204
- "Uninstall Page" on page 206

Macros Page

The Macros property page, shown below, displays the macros that have been defined for use with the Application object.



These macros, referred to as Application object macros, are simply variable names that are associated with values, typically string values. For example, when you create an Application object using an .AOT or .AXT file, a SOURCE_PATH macro is automatically added to the list. This macro defines the location of the source files from which the application will be installed. Many times, you must enter the source path when defining other properties for the Application object. By creating a SOURCE_PATH macro, you only need to enter %SOURCE_PATH% rather than the actual source path.

You can use Application object macros when defining the following Application object properties:

- Path to Executable (Run Options > Applications page)
- Command Line (Run Options > Applications page)
- Working Directory (Run Options > Applications page)
- Drive Mapping Path (Common > Drives/Ports page)
- Capture Port Path (Common > Drives/Ports page)
- Key, Value Name, and Value Data strings (Distribution Options > Registry Settings page)

- Section, Value Name, and Value Data strings (Distribution Options > INI Settings page)
- Source File, Target File, Directory Name (Distribution Options > Application Files page)
- Find File, Find Next File, Add string (Distribution Options > Text Files page)
- All properties (Distribution Options > Icons/Shortcuts page)

IMPORTANT: When using a macro for any of these properties, you must enclose the macro name in % characters (for example, %SOURCE_PATH%).

Macros

The Macros list displays all macros that have been defined for use with the Application object.

File

Click File > Find to search for specific information in the Name, Value, and Prompt fields.

Click File > Import to import macros from another Application object's .AOT or .AXT file.

Add

Use the Add button to define an Application object macro and add it to the Macros list. You can add either a String Value macro or a Prompted macro.

String Value

A String Value macro is used to create a variable that has a string value associated with it. This lets you use the variable in multiple locations while defining the value in one location.

To create a String Value macro, click Add > String Value to display the Edit String Value dialog box. In the Value Name box, type a name for the variable > type the value for the variable in the Value Data box. Click OK to add the macro to the Macros list.

You can place macros within macros. In the following example, a special Windows macro, *WINDISK, is used when defining the TARGET_PATH macro:

```
Value Name: TARGET_PATH
Value Data: %*WINDISK%\Program Files
```

When embedding another macro in the string, you must enclose the macro in % characters, as shown in the above example. You can embed special Windows macros, NDS attribute macros, environment variable macros, login script macros, and other Application object macros. For information about these types of macros, see Chapter 13, “Macros,” on page 217.

Prompted

A Prompted macro is used to create a macro that prompts the user to accept the default value (either a drive or string) or enter a new value.

To create a Prompted Drive macro, click Add > Prompted > Drive to display the Prompted Drive Macros dialog box. Fill in the fields as described below:

- **Macro Name:** Give the macro any name you want. Do not use spaces.
- **Prompt Text:** Enter the text you want presented to users. For example, "This application will be installed to your C: drive. Enter a different drive letter if desired:".
- **Default Value:** Select the drive that will be used if the user does not enter another drive letter.
- **Minimum Disk Space in MB:** Enter the minimum amount of free disk space required to install the application.

To create a Prompted String macro, click Add > Prompted > String to display the Prompted String Macros dialog box. Fill in the fields as described below:

- **Macro Name:** Give the macro any name you want. Do not use spaces.
- **Prompt Text:** Enter the text you want presented to users.
- **Default Value:** Select the value that will be used if the user does not enter another value in response to the prompt text.
- **Minimum Disk Space in MB:** Enter the minimum amount of free disk space required to install the application.
- **Maximum String Length in Chars:** Enter the maximum number of characters the user can enter as a response to the prompt.

Modify

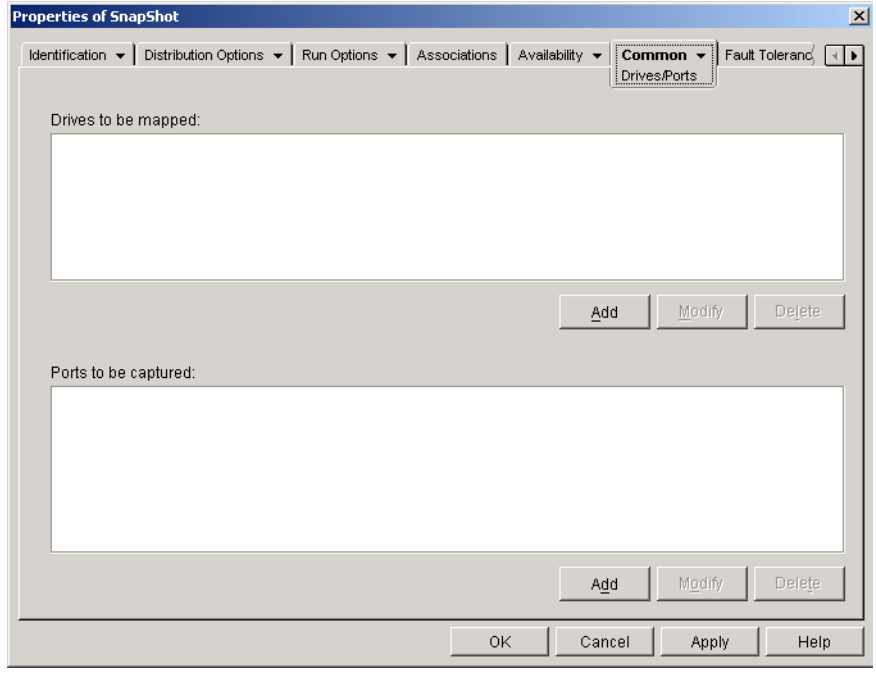
Select the macro whose information you want to modify > click Modify.

Delete

Select the macro you want to delete > click Delete.

Drives/Ports Page

The Drives/Ports property page, shown below, determines the drive mappings and port captures for the application. When a user launches the application (through the Application object), Application Launcher/Explorer establishes the drive mappings and port captures before launching the application.



For example, suppose you've set up a database application to run from drive W:. To ensure that W is mapped to the location of the application, you could map drive W: to the server, volume, and directory where the application exists. When Application Launcher/Explorer runs the application, it establishes the drive mapping according to the conditions defined when setting up the drive mapping.

Or, maybe users need access to a network location to store files created with a word processor. If you map a drive, the drive mapping becomes active as soon as the application runs.

By default, drive mappings and port captures are not released when the user closes the application. For Application Launcher/Explorer to release the drives and ports, you need to select the Clean Up Network Resources option (Run Options > Environment page). With this option selected, Application Launcher/Explorer releases any drives it mapped and ports it captured as long as the drives and ports are not being used by another application. If several applications use the same drive or port, then the drive mappings and port captures aren't cleaned up until the last application terminates.

Drives to Be Mapped

The Drives to Be Mapped list displays all drives that will be mapped before the application is launched.

Add

Click Add to display the Drive to Be Mapped dialog box. Fill in the following fields:

- **Root:** Check this option to treat the path as the root of the drive.
- **Option:** Select the type of drive to map. The available types are:
  - DRIVE: Select this option to map to a drive letter (A through Z).
  - S1: Select this option to assign the drive to the first available search drive
  - S2: Select this option to assign the drive to the last search drive.
- **Drive:** Assign a drive letter. The options are drives A through Z, or the next available drive.
- **Path:** Enter, or browse and select, the path you want mapped to the selected drive. If you enter the path, use the following syntax:

  *server**volume*:*path*
  *server**volume**path*
  volume_object_name:*path*
  directory_map_object_name:*path*
  driveletter:*path*

  You can also use a macro in this field. For information about macros, see Chapter 13, "Macros," on page 217.

- **Map If Drive Mapping Doesn't Exist or Same as Existing:** Select this option if you want Application Launcher/Explorer to map the drive if 1) it doesn't exist already or 2) it is the same as an existing drive mapping.
- **Overwrite any Existing Drive Mapping:** Select this option if you want Application Launcher/Explorer to replace any mapping that uses the requested drive letter. For example, if S: is mapped to \\SERVER1\SYS\PUBLIC and you've specified the new S: drive mapping as \\SERVER1\APPS\TIMECARD, the end result is that drive S: is mapped to \\SERVER1\APPS\TIMECARD.

  To identify this option in the Drives to Be Mapped list, the word FORCE is added to the mapping information.
- **Use the Existing Drive Mapping If Already Mapped:** Select this option to use the existing drive mapping even if it is not the same as the path requested. For example, if S: is mapped to \\SERVER1\SYS\PUBLIC and you've specified the new S: drive mapping as \\SERVER1\APPS\TIMECARD, the end result is that drive S: is still mapped to \\SERVER1\SYS\PUBLIC.

  To identify this option in the Drives to Be Mapped list, the word REUSE is added to the mapping information.

Modify

Select a drive mapping in the Drives to Be Mapped list > click Modify to display the Drive to be Mapped dialog box. Change the drive mapping information as needed. See the field descriptions above or click the Help button in the dialog box.

Delete

Select a drive mapping in the Drives to Be Mapped list > click Delete to remove the mapped drive.

Ports to Be Captured

The Ports to Be Captured list displays all ports that will be captured before the application is launched.

Add

Click Add to display the Ports to Be Captured dialog box. Fill in the following fields:

- **Port:** Select an LPT port to assign to the printer or queue.
- **Printer or Queue:** Browse to select a printer or queue.

  You can also use a macro in this field. For information about macros, see Chapter 13, “Macros,” on page 217.
- **Capture Flags:** You can use this option to override notification, banner, and form feed settings defined on the user's workstation. Whenever a user prints from the application, the settings you defined will be used.

  In the Capture Flags box, select the Override Workstation Setting box for the corresponding Capture setting you want to override. After you do this, the corresponding Capture setting becomes active. Select the setting to turn it on. Leave the setting unchecked to turn it off.

  For example, if you select the Override Workstation Setting box for Notify and also select the Capture Setting box for Notify, when the user prints a document, the Novell Client displays a Notify alert panel to indicate the completion of the print job, no matter which Capture settings are on the user's workstation. However, if you select the Override Workstation Setting box for Notify but do not select the Capture Setting box for Notify, when the user prints a document, the Novell Client does not display a Notify alert panel to indicate the completion of the print job, no matter which Capture settings are used on the user's workstation.

Modify

Select a port capture in the Ports to Be Captured list > click Modify to display the Ports to be Captured dialog box. Change the capture information as needed. See the field descriptions above or click the Help button in the dialog box.

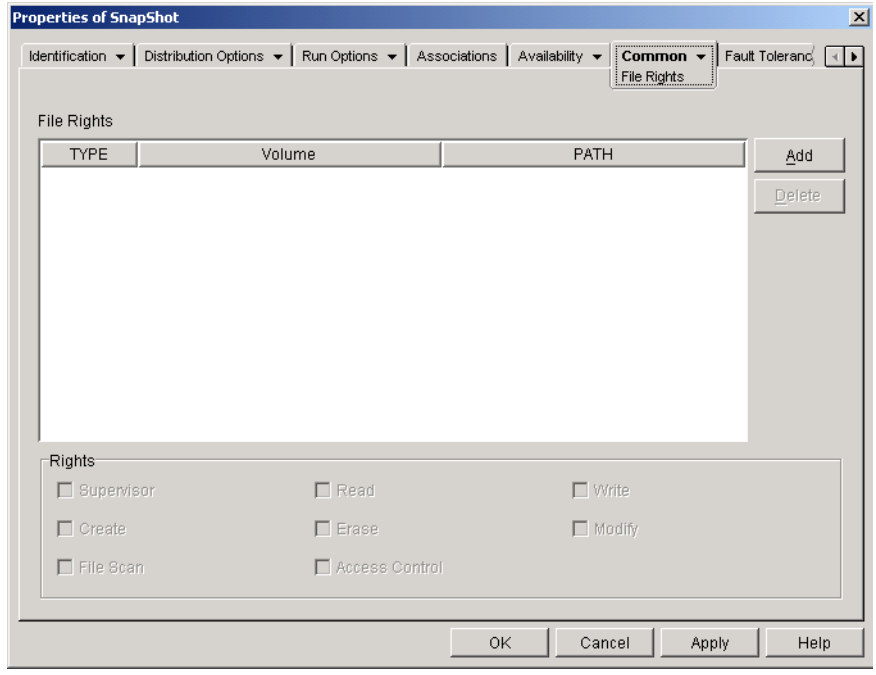
Delete

Select a port capture in the Ports to Be Captured list > click Delete to remove the captured port.

File Rights Page

The File Rights property page, shown below, specifies the file, directory, and volume rights a user must have to run the application. A user receives these rights when his or her User object is associated with the Application object, or when a Group, Organizational Unit, Organization, or Country object of which the user is a member is associated with the Application object. The rights are

removed from the user when the object is no longer associated with the Application object.



The file rights you set up are not dependent on whether the user is actually using the application. The user has continual file rights from the time the Application object is associated with the user until the time the association is removed.

If two separate Application objects give file rights to the same file, directory, or volume and the user is unassociated with only one of them, then the user loses all rights even though the user might still be associated with the other Application object.

File Rights

The File Rights list displays all files, directories, and volumes to which rights will be given. When you select a file, directory, or volume, the rights that have been granted are displayed in the Rights box. You can grant or remove rights by checking or unchecking the appropriate boxes.

Add

Click Add > browse for and select the file, directory, or volume you want to add > click OK to display it in the list.

Delete

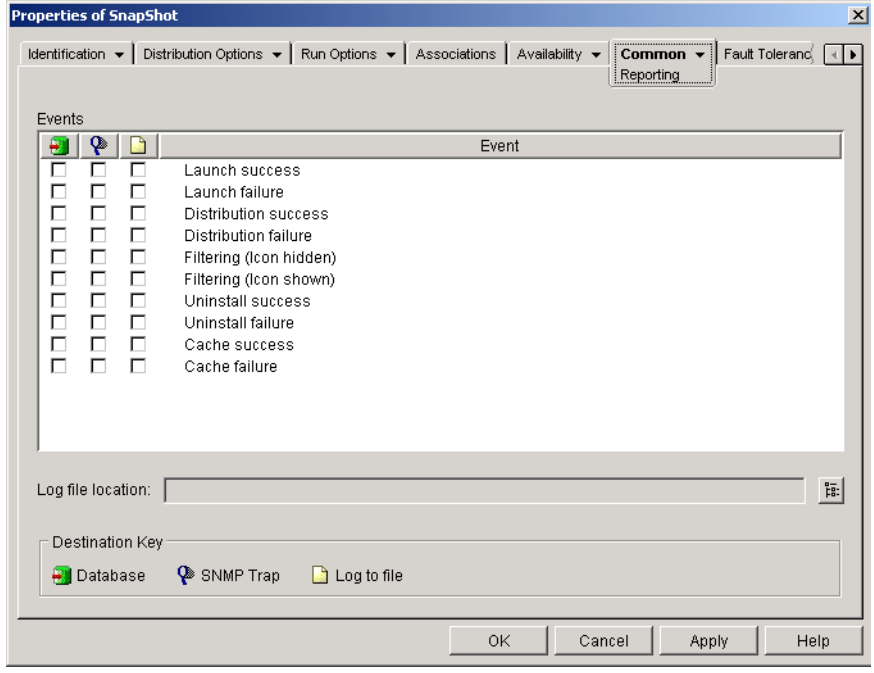
Select the file, directory, or volume you want to delete from the list > click Delete. Any users associated with the Application object will lose rights to the file, directory, or volume.

Rights

Select the file, directory, or volume (in the File Rights list) for which you are establishing users' rights > check the appropriate boxes.

Reporting Page

The Reporting property page, shown below, specifies the application events that you want Application Launcher/Explorer to report on and lets you specify the method of reporting.



Reports contain the following information: event type, date and time, user distinguished name, workstation distinguished name, workstation address, application distinguished name, application global unique identifier (GUID), application version string, and event strings.

For information about setting up reporting, see Chapter 9, “Reporting on Application Management Events,” on page 97.

Events

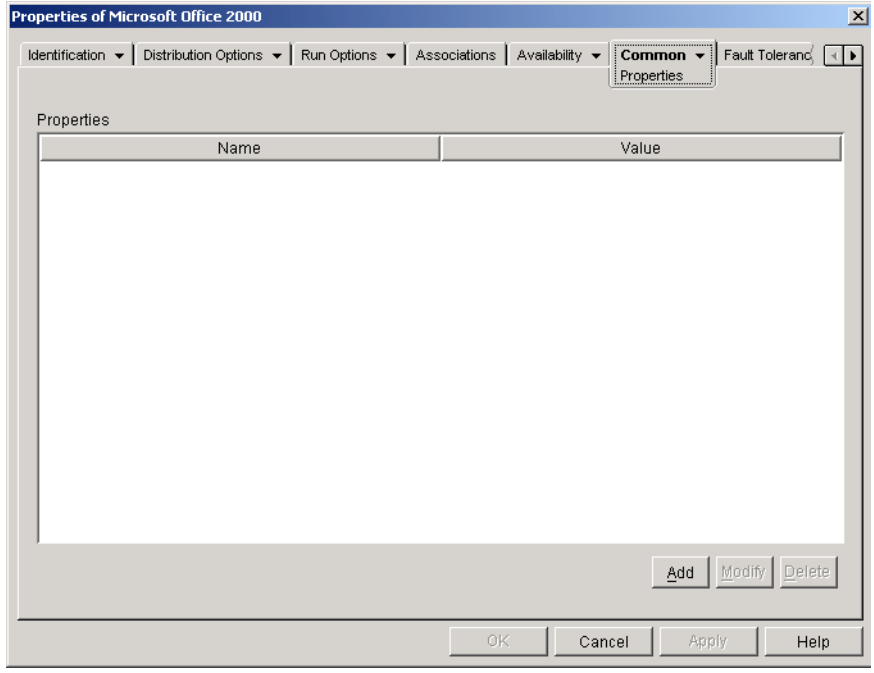
Select the events (distribution, launch, filtering, and uninstallation) that you want reported and the destination where you want the events sent. You can send events to a database, to a management console via an SNMP trap, or to a text log file. If necessary, you can send events to multiple destinations (for example, both a database and a text log file).

Log File Location

If you selected a log file as the destination for any of the event reports, enter (or browse to and select) the location and name for the log file.

Properties Page (.MSI Application Object Only)

The Properties property page, shown below, lets you override the public property values contained in the .MSI package.



The .MSI package contains the property values that were defined during the administrative installation of the application. These properties determine the way the Microsoft Windows Installer installs the application to the user's workstation. In some cases, you may want to change one or more of the properties. For example, a property may contain the default location for a user's work files. By adding the property to the Properties list and changing the property's value, you can override the default location defined in the .MSI package.

Add

To add a property to the Properties list, click Add to display the Select Property to Add dialog box. In the Value Name field, select the property whose value you want to override > enter the new value in the Value Data field > click OK to add the property to the Properties list.

Modify

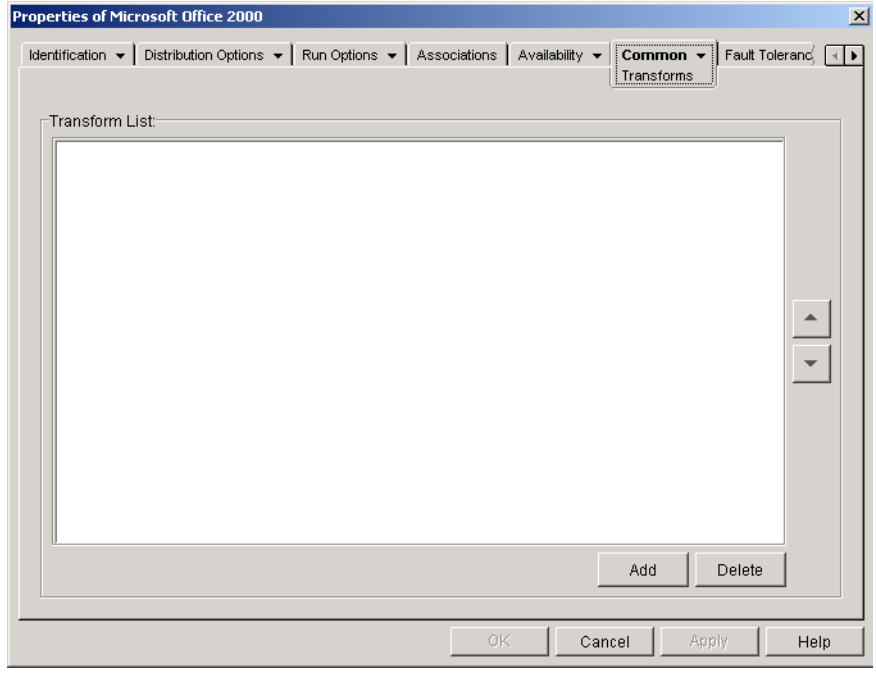
To modify a property that is in the Properties list, select the property > click Modify to display the Edit String Value dialog box > modify the value data > click OK.

Delete

To delete a property from the Properties list, select the property > click Delete. Deleting the property causes future installations of the application to use the property value defined in the .MSI package.

Transforms Page (.MSI Application Object Only)

The Transforms property page lists the transforms that Microsoft Windows Installer will apply to the .MSI package during distribution. A transform adds, deletes, or changes properties in the .MSI package to enable customizing of the installation for different groups of users.



Transforms are applied in the order they appear in the Transforms List. If two transforms modify the same package property, the property retains the value applied by the last transform.

For information about creating transforms for your .MSI package, see the documentation you received with the software application.

Add

Click Add > browse to and select the transform file (.MST file) you want added to the Transform List > click OK to add it to the list. If the transform file is in the same location as the .MSI package, you can simply enter the filename in the Transform Path field.

Delete

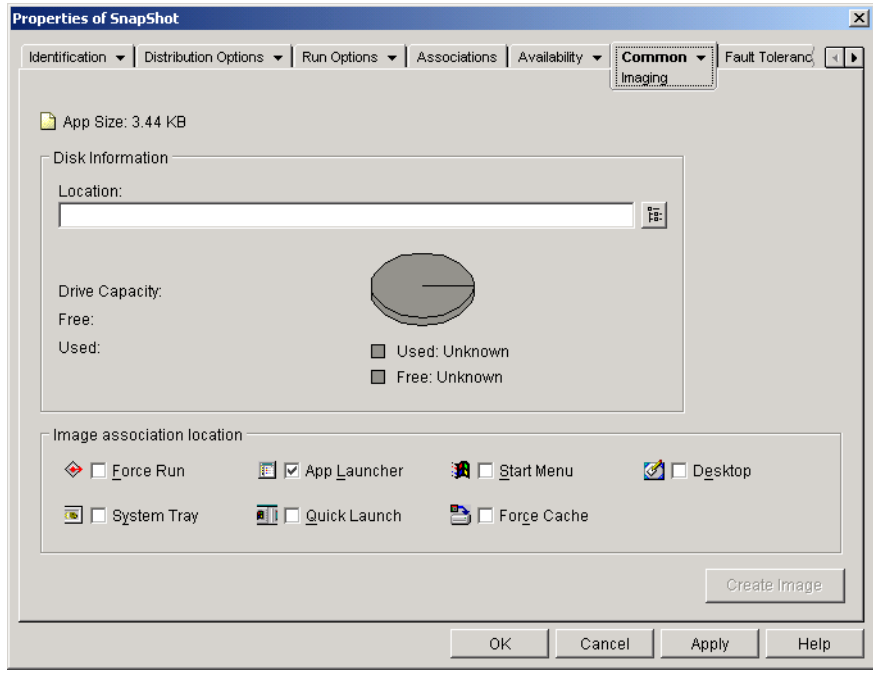
Select a transform from the Transforms List > click Delete to remove it from the list.

Up-Arrow and Down-Arrow

Select a transform from the Transforms List > click the up-arrow to move the transform up in the list or click the down-arrow to move it down in the list. Application Launcher/Explorer applies the transforms in the order they are listed, from top to bottom.

Imaging Page

The Imaging property page, shown below, lets you create an image file for the Application object. You can use the image file as an add-on image to a base image file.



App Size

This field displays the size of the Application object. This includes all Application object information (.AOT or .AXT file) as well as application source files (.FIL files).

Disk Information

These fields let you specify a location and filename for the image and ensure that the disk to which you are saving the image has sufficient free space.

Location

Specify the location and filename for the image. You may want to save it in the same location as the base image file. The filename does not have to have a particular filename extension.

Drive Capacity, Free, and Used

Once you've entered a location for the file, these fields display the amount of space on the disk, the amount of available free disk space, and the amount of

used disk space. Verify that the Application object will fit in the amount of free disk space.

Image Association Location

Specify the association characteristics for the Application object. These will be enforced when the image is applied to the workstation.

Force Run

Runs the Application object as soon as Application Launcher/Explorer or Workstation Helper starts and the Application object is available.

You can use the Force Run option in conjunction with several other settings to achieve unique behaviors. For example, if you use the Force Run option with the Run Application Once option (Run Options > Applications page), as soon as the Application object is distributed it will run one time and then be removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a pre-determined time. If so, select Force Run, select the Run Application Once option on the Application page (Run Options tab), and define a schedule using the Schedule page (Availability tab).

If you want to force run several applications in a specific order, mark each of them as Force Run. Then give each Application object a numeric order by using the Order Icon Display option on the Icon page (Identification tab).

App Launcher

Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu

If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be added to the top of the Start menu, unless you assign the Application object to a folder and use the folder structure on the Start menu. See the Application object's Folders page (Identification tab).

Desktop

If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows desktop.

System Tray

If the workstation is running Application Explorer, this option displays the Application object's icon in the Windows system tray.

Quick Launch

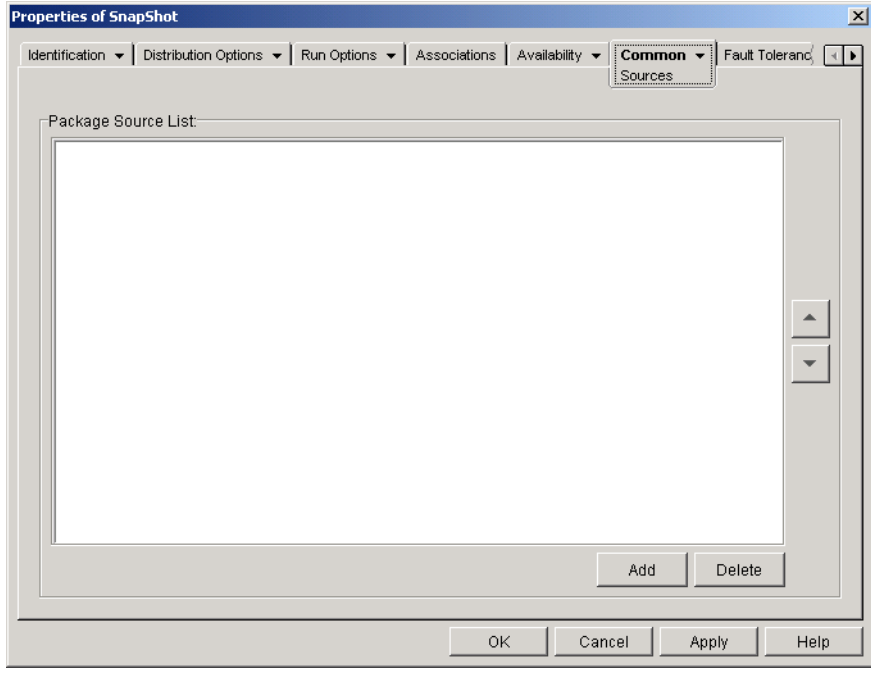
Displays the Application object's icon on the Windows Quick Launch toolbar.

Force Cache

Forces the application source files and other files required for installation to be copied to the workstation's cache directory. The user can then install or repair the application while disconnected from NDS. The files are compressed to save space on the workstation's local drive.

Sources Page

The Sources property page, shown below, specifies additional network locations that contain installation packages from which Application Launcher/Explorer can distribute the application. If the Application object was created from a .AOT or .AXT file, the installation packages must be snAppShot packages (.AOT, .AXT, and .FIL files). If the Application object was created from a .MSI file, the installation packages must be Microsoft Windows Installer packages (.MSI files). The source list is used only if the original source location is unavailable, in which case Application Launcher/Explorer begins with the first location in the list and continues down the list until an accessible package is found.



Before you can define additional source locations, you need to make sure to copy the original package to the new source locations. For snAppShot packages, copy the .AOT, .AXT, .FIL, and .TXT files. For Windows Installer packages, copy the administrative installation folder and all subfolders.

With Application objects that use a .MSI file rather than a .AOT or .AXT file, this source list can be extremely important. Because Windows Installer supports on-demand installations (installation of files as the user requests certain features or components), the source list can provide source resiliency for installations that occur after the initial installation. If the original installation source is no longer available and no source list has been established, the on-demand installation will fail. However, if you create additional source locations and define them in the source list, the on-demand installation will have a better chance of succeeding.

IMPORTANT: If you have used the SOURCE_PATH macro (Common > Macros page) in any of the Application object's property fields, you should verify that you used all uppercase letters and no lowercase letters. If the original source location is referenced through the SOURCE_PATH macro and Application Launcher/ Explorer cannot access the location, it will only use the package sources in this list if the SOURCE_PATH macro is all uppercase.

Package Source List

The Package Source List displays the additional package source locations that have been defined for the application. When the application is distributed to a workstation, Application Launcher/Explorer will use these locations, in the order listed, if the original source location is unavailable.

When distributing applications that use the Windows Installer, Application Launcher/Explorer passes the source list to the Windows Installer, which then uses the list for the locations from which to install the application files.

Add

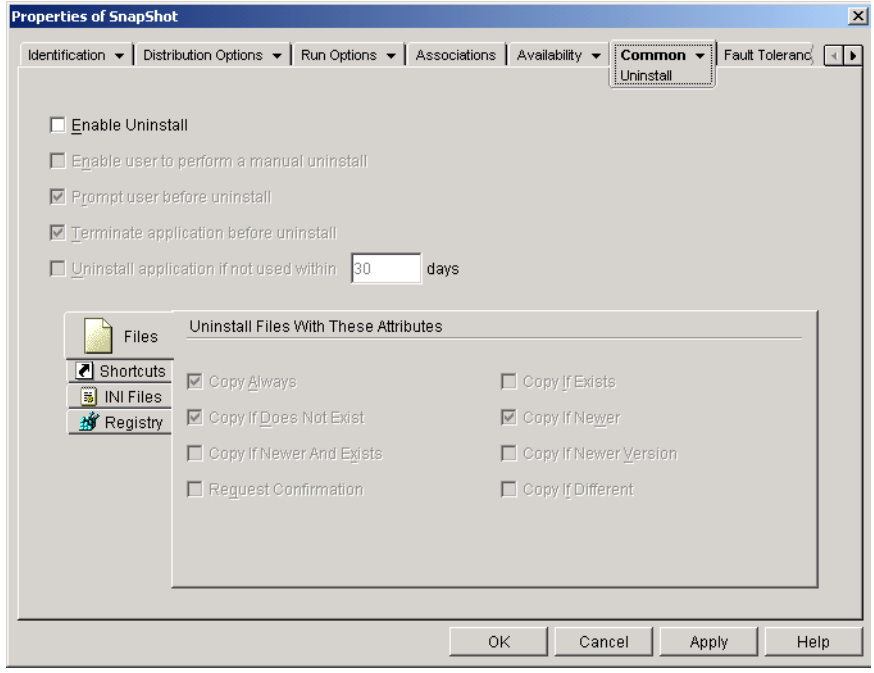
To add a source location to the list, click Add > browse to and select the directory that contains the installation package > click OK.

Delete

To remove a source location from the list, select the location > click Delete.

Uninstall Page

The Uninstall property page, shown below, enables the application to be uninstalled and determines the behaviors associated with uninstalling the application.



Enable Uninstall

Check this option to enable the application to be removed from the workstation. The other uninstall options are available only if this option is enabled.

Enable User to Perform a Manual Uninstall

Check this option to enable users to remove the application from their workstations. If this option is not enabled, only you or other administrators can remove applications.

Prompt User Before Uninstall

Check this option to prompt users before removing the application from their workstations. If users answer "No" to the prompt, the application will not be removed.

Terminate Application Before Uninstall

Check this option to ensure that Application Launcher/Explorer terminates the application (if the application is running) before it begins uninstalling the application files.

Uninstall Applications If Not Used Within *XX* Days

Check this option to automatically remove the application if the user has not run it within the specified number of days (the default is 30).

Files

Click Files > check the attributes you want to use to determine whether or not an application file is removed. The attributes correspond to the attributes listed on the Application Files page (Distribution Options tab). By default, the options are configured (Copy Always, Copy if Does Not Exist, Copy if Newer) to ensure that a file is removed only if it was installed when the application was installed and not at an earlier time by another application that also needs the file.

Shortcuts

Click Shortcuts > check the attributes you want to use to determine whether or not the application's shortcuts are removed. The attributes correspond to the attributes listed on the Shortcuts/Icons page (Distribution Options tab).

INI Files

Click INI Files > check the attributes you want to use to determine whether or not an INI setting is removed. The attributes correspond to the attributes listed on the INI Settings page (Distribution Options tab). By default, the options are configured (Create Always, Create if Exists, Create or Add to Existing Section) to ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that also needs the setting.

Registry

Click Registry > check the attributes you want to use to determine whether or not a registry setting is removed. The attributes correspond to the attributes listed on the Registry Settings page (Distribution Options tab). By default, the options are configured (Create Always, Create if Exists) to ensure that a

setting is removed only if it was created when the application was installed and not at an earlier time by another application that still needs the setting.

Uninstall Keys/Values From These Registry Hives

This option allows you to select two specific registry hives, HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER, to include or exclude when removing the registry settings that were added by Application Launcher/Explorer during distribution of the application to the workstation. The default settings cause Application Launcher/Explorer to remove settings from the HKEY_CURRENT_USER hive but not from the HKEY_LOCAL_MACHINE hive.

The purpose of this option is to help you ensure that no Windows system settings are removed during the uninstall. This is particularly important if you used snAppShot to capture the application's installation. When snAppShot captures an application's installation, it includes all settings that are changed during the installation. If the application's installation program causes the workstation to reboot during the installation process, these changes can include not only settings that apply to the application but also settings that apply to the Windows system. Removing Windows system settings may cause the workstation to encounter problems when starting up. It is strongly recommended that you remove these settings only if you are positive that they will not affect the workstation's ability to function properly.

NOTE: If you want to use this option, you should examine the registry settings that will be removed from the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER hives and delete any settings that apply to anything other than the application. The registry settings are displayed on the Registry Settings page (Distribution Options tab).

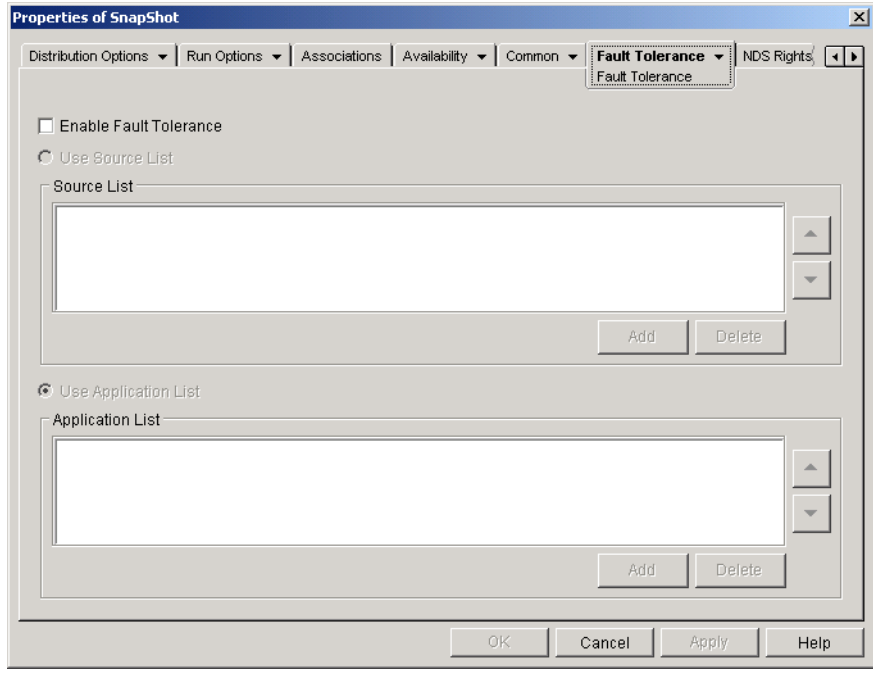
Fault Tolerance Tab

The Fault Tolerance tab includes the following pages to help you balance the workload required to host an application between multiple servers and provide access to backup applications in case of server failure:

- "Fault Tolerance Page" on page 210
- "Load Balancing Page" on page 212
- "Site List Page" on page 214

Fault Tolerance Page

The Fault Tolerance property page, shown below, specifies the installation package sources or Application objects that Application Launcher/Explorer will use as backups if this Application object becomes unavailable. Application Launcher/Explorer tries the package sources or Application objects in the order shown in the Source List or Application List, from top to bottom.



We recommend you use fault tolerance, rather than load balancing (Fault Tolerance > Load Balancing page) when servers are located at various sites throughout a large wide area network. However, the servers must reside in the same NDS tree. You can implement both methods if necessary. Application Launcher/Explorer will try load balancing first, then fault tolerance.

IMPORTANT: The Fault Tolerance page does not apply to Application objects that use Microsoft Windows Installer (.MSI) packages. For a .MSI Application object, you can provide source resiliency (fault tolerance) through the use of additional source locations for the installation package (Common > Sources page).

Enable Fault Tolerance

Check this option to enable fault tolerance. The other fault tolerance options are available only if this option is enabled.

Use Source List

Select this option to have Application Launcher/Explorer use a list of Package sources as the backup. You must have already created at least one installation package source (Common > Sources page).

Add

Click Add > browse to and select a package source > click OK to add it to the Source List.

Delete

Select a package source from the Source List > click Delete to remove it from the Source List.

Up-Arrow and Down-Arrow

Select a package source from the Source List > click the up-arrow to move the source up in the list or click the down-arrow to move it down in the list. Application Launcher/Explorer tries the sources in the order they are listed, from top to bottom.

Use Application List

Select this option to have Application Launcher/Explorer use a list of Application objects as the backup. You must have already created additional Application objects for the application, with the application files being stored on a different server or volume than this application. If the application is too busy or is not available, then Application Launcher/Explorer tries each Application object in the specified order.

Add

Click Add > browse to and select an Application object > click OK to add it to the Application List.

Delete

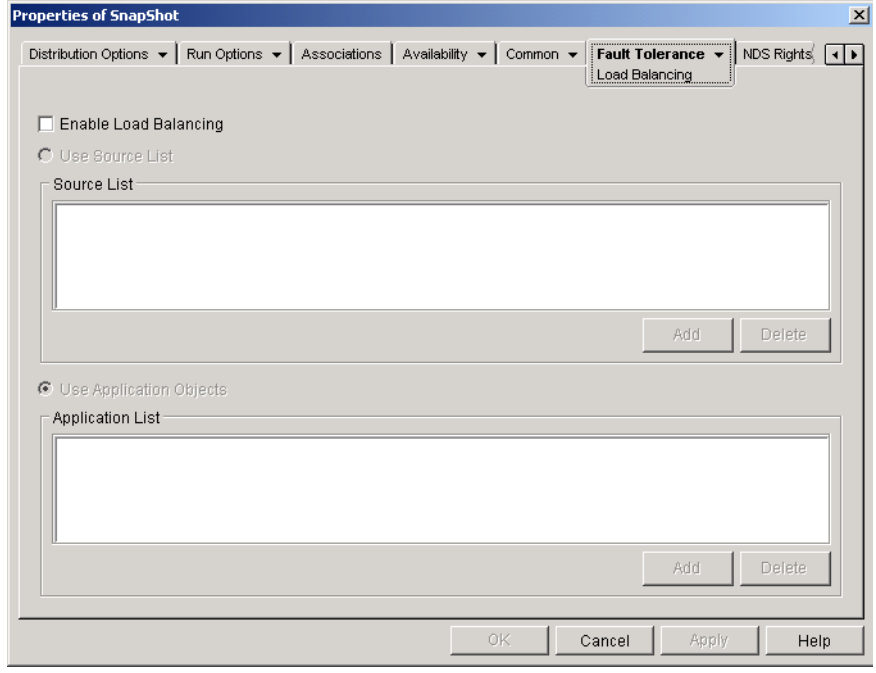
Select an Application object from the Application List > click Delete to remove it from the Application List.

Up-Arrow and Down-Arrow

Select an Application object from the Source List > click the up-arrow to move the Application object up in the list or click the down-arrow to move it down in the list. Application Launcher/Explorer tries the Application objects in the order they are listed, from top to bottom.

Load Balancing Page

The Load Balancing property page, shown below, lets you balance the workload required to support the application. With load balancing enabled, Application Launcher/Explorer will pull the application files from all servers offering the application, ensuring a more balanced workload across servers.



To use load balancing, you must first have multiple installation package sources or Application objects defined for the application, with the application

source files located on different servers. When a user launches the application, Application Launcher/Explorer selects one of the package sources or Application objects to use. If the package source or Application object is unavailable, Application Launcher/Explorer selects another one.

We recommend you use load balancing when all the servers for the Application objects are located at a single site. If the servers are located at various sites throughout a large wide area network, we recommend you use fault tolerance (Fault Tolerance > Fault Tolerance page). You can implement both methods if necessary. Application Launcher/Explorer will try load balancing first, then fault tolerance.

Enable Load Balancing

Check this option to enable load balancing. The other load balancing options are available only if this option is enabled.

IMPORTANT: For Application objects that use Microsoft Windows Installer (.MSI) packages, the Enable Load Balancing option is the only option that is enabled on this page; the Use Source List and Use Application List options are disabled. To enable load balancing for a .MSI Application object, check this option and then use the Application object's Sources page (Common tab) to define the list of source locations where you've created additional installation packages for the application. Application Launcher/Explorer will randomly select source locations from the list to balance the workload.

Use Source List

Select this option to have Application Launcher/Explorer use a list of package sources for load balancing. You must have already created at least one package source (Common > Sources page).

Add

Click Add > browse to and select a package source > click OK to add it to the Source List.

Delete

Select a package source from the Source List > click Delete to remove it from the Source List.

Use Application List

Select this option to have Application Launcher/Explorer use a list of Application objects for load balancing. You must have already created additional Application objects for the application, with the application files being stored on a different server or volume than this application.

Add

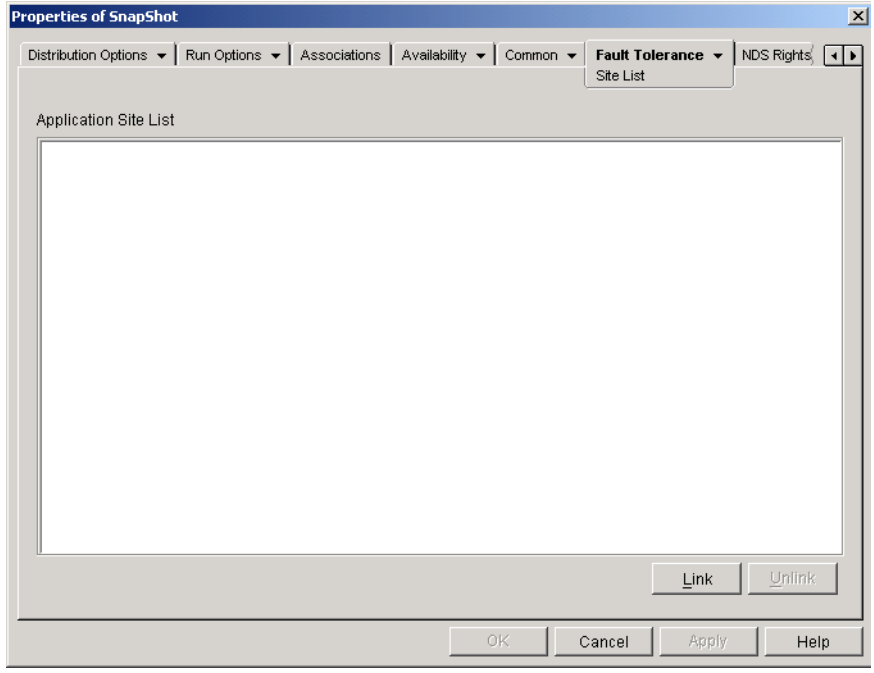
Click Add > browse to and select an Application object > click OK to add it to the Application List.

Delete

Select an Application object from the Application List > click Delete to remove it from the Application List.

Site List Page

The Site List property page, shown below, lists the other Application objects to which this Application object is linked. By linking identical (or similar) Application objects whose application source files are located on servers at different sites, you can create an application site list that gives users who travel from site to site fast access to their applications while reducing WAN traffic and associated costs.



For example, users at two different sites run a spreadsheet application from servers that are local to their sites. Different Application objects are used to distribute and launch the spreadsheet application from the two servers. If you link the two Application objects, any Site1 users who travel to Site2 will be able to run the application from the Site2 server the same way they would run it from their own Site1 server. The same would be true for Site2 users who travel to Site1.

You can link to one other Application object only. However, when you do so, the first Application object is automatically linked to any other Application objects that the second Application object is linked to. For example, you have three identical Application objects (App1, App2, and App3) at three different sites. You link App1 to App2, which creates the following site lists for each of the Application objects:

```
App1 linked to: App2
App2 linked to: App1
App3 linked to: (nothing)
```

Because you can link an Application object to only one other Application object, you now need to link App3 to either App1 or App2. Doing so creates the following site list for each of the Application objects:

```
App1 linked to: App2, App3
App2 linked to: App1, App3
App3 linked to: App1, App2
```

Link

Click Link > browse for and select the Application object you want to link to > click OK to add it to the Application Site List. The list will also include any other Application objects that the selected Application object is linked to.

Unlink

Click Unlink to remove the Application object's link to the Application objects displayed in the Application Site List.

13 Macros

You can use macros with many of the properties available on the Application object pages in ConsoleOne™. You can use any of the five types of macros listed below.

- Application object macros
- Special Windows* macros
- Login script macros
- NDS® attribute macros
- Environment variable macros.

Each macro type gives you various abilities. For example, Application object macros are basically variables to which you can assign values. By using an Application object macro, you have the flexibility to change the macro's value in one location and have it automatically changed anywhere you have used the macro. Special Windows macros, on the other hand, let you specify Windows locations, such as the Windows temporary directory, by specifying a variable name (for example, %*WINTEMP%) that exists on the workstation and defines the location on that workstation. This enables you to specify locations that may vary from workstation to workstation.

The following sections provide information about the five types of macros supported by Application objects, the precedence order, and the Application object properties where macros can be used.

- "Application Object Macros" on page 218
- "Special Windows Macros" on page 220
- "Login Script Macros" on page 225
- "NDS Attribute Macros" on page 228
- "Environment Variable Macros" on page 230
- "Macro Precedence" on page 231
- "Application Object Properties Where Macros Can Be Used" on page 231

Application Object Macros

An Application object macro is defined on the Application object's Macros page and can only be used with that Application object. The macro is simply a variable that has a value associated with it, as shown in the following two examples:

Variable	Value
SOURCE_PATH	\\server1\sys\public\zen\word
TARGET_PATH	c:\program files\word

In the examples, SOURCE_PATH defines the network location from which Novell® Application Launcher™/Explorer will distribute the application to the user's workstation, while TARGET_PATH defines the workstation location to which Application Launcher/Explorer will copy the files.

You can also create a macro that prompts the user to accept a default value you've assigned or to enter a different value. For example, you could use a macro that prompts the user with a default installation drive (such as C:) but lets the user change it to another drive (such as D:).

The following sections provide information about nesting other macros in Application object macros and instructions to help you define Application object macros:

- "Nesting Macros" on page 218
- "Defining an Application Object Macro" on page 219

Nesting Macros

When defining an Application object macro, you can embed other macros in the definition. In the following example, a special Windows macro, *WINDISK, is used when defining the TARGET_PATH macro:

```
%*WINDISK%\Program Files
```

Any time you use the TARGET_PATH macro in an Application object property field, Application Launcher/Explorer would substitute the current workstation's Windows drive letter for %*WINDISK% (for example, C:\TEMP).

When embedding another macro in the string, you must enclose the macro in % characters, as shown in the above example. You can embed special Windows macros, NDS attribute macros, environment variable macros, login script macros, and other Application object macros.

Defining an Application Object Macro

1. In ConsoleOne, right-click the Application object for which you want to define the macro > click Properties.
2. Click the Common tab > click Macros to display the Macros page.
3. To define a macro consisting of a variable name and a value (a String Value macro), continue with "Defining a String Value (Non-Prompted) Macro" on page 219.

   or

   To define a macro that prompts the user to accept a default value or enter a different value, skip to "Defining a Prompted Macro" on page 219.

Defining a String Value (Non-Prompted) Macro

1. On the Macros page, click Add > click String Value to display the Edit String Value dialog box.
2. In the Value Name box, type a name for the macro variable (for example, TARGET_PATH). Do not use spaces in the name.
3. In the Value Data box, type the value for the macro variable (for example, c:\program files\word).
4. Click OK to add the macro to the Macros list.

For information about using the macro in a valid Application object property, see "Application Object Properties Where Macros Can Be Used" on page 231.

Defining a Prompted Macro

1. On the Macros page, click Add > click Prompted > click Drive to create a macro that prompts the user for a drive or click String to create a macro that prompts the user for text input.
2. Fill in the following fields:

   Macro Name: Enter any name that describes the macro. Do not use spaces in the name.

Prompt Text: Enter the text you want presented to the user. For example, if you are prompting for an installation drive you could enter the following: "This application will be installed to your C: drive. Enter a different drive letter if desired."

Default Value: For a prompted drive macro, select the drive that will be displayed as the default drive. For a prompted string macro, enter the text that will be displayed as the default text.

Minimum Disk Space in MB: Enter the minimum amount of free disk space required. The user will not be allowed to install to any drive that does not meet the minimum disk space requirement.

Maximum String Length in Chars (Prompted String Macro Only): Enter the maximum number of characters the user can enter as a response to a prompted string macro.

3 Click OK to add the macro to the Macros list.

For information about using the macro in a valid Application object property, see "Application Object Properties Where Macros Can Be Used" on page 231.

Special Windows Macros

A special Windows macro is one that defines Windows 95/98 and Windows NT*/2000 directories. The typical paths listed below are based on default installations and may or may not match your specific setup. On Windows 95/98 workstations, macros behave differently if User Profiles are enabled.

Suppose that you have installed Windows to drive D: (for example, D:\WINDOWS). However, an application installation expects Windows to be on drive C: (for example, C:\WINDOWS). Using the *WinDisk macro, you can substitute drive C: with the macro for the files that require it.

The following two sections describe the special Windows macros you can use.

- "Windows 95/98 and Windows NT/2000 Macros" on page 221
- "Windows 2000 Macros" on page 224

For information about using the macros in a valid Application object property, see "Application Object Properties Where Macros Can Be Used" on page 231.

Windows 95/98 and Windows NT/2000 Macros

The following macros apply to Windows 95/98 and Windows NT/2000 unless otherwise noted.

IMPORTANT: The asterisk character (*) is a required syntax for these macros. Don't confuse these asterisk characters with the Novell trademark asterisk. In addition, when using the macro in one of the Application object's property fields, you must enclose it in % characters (for example, %*WinDisk%).

Macro	Description
%*AppData %	File system directory that serves as a common repository for application specific data. Typically C:\WINNT\PROFILES*user name*\APPLICATION DATA. Used only by Windows NT version 4.
%*CommonDesktop%	File system directory that contains files and folders that appear on the desktop for all users. Typically C:\WINNT\PROFILES\ALL USERS\DESKTOP. If not available, the *DESKTOP value will be used.
%*CommonPrograms%	File system directory that contains the directories for the common program groups that appear on the Start menu for all users. Typically C:\WINNT\PROFILES\ALL USERS\START MENU\PROGRAMS. If not available, the *PROGRAMS value will be used.
%*CommonStartMenu%	File system directory that contains the programs and folders that appear on the Start menu for all users. Typically C:\WINNT\PROFILES\ALL USERS\START MENU. If not available, the *STARTMENU value will be used.
%*CommonStartup%	File system directory that contains the programs that appear in the Startup folder for all users. The system starts these programs whenever any user logs on to Windows NT or starts Windows 95/98. Typically this directory is C:\WINNT\PROFILES\ALL USERS\START MENU\PROGRAMS\STARTUP. If not available, the *STARTUP value will be used.

Macro	Description
%*CommonWinDesktop%	Windows NT common desktop directory (C:\WINNT\PROFILES\ALL USERS\DESKTOP). This is a Novell Application Launcher 2.0 macro that is allowed for backward compatibility.
%*Cookies%	Files system directory that contains the user's cookies. Typically C:\WINDOWS\COOKIES or C:\WINNT\PROFILES*username*\COOKIES.
%*Desktop%	File system directory used to physically store file objects on the desktop (not to be confused with the desktop folder itself). Typically this directory is C:\WINDOWS\DESKTOP or C:\WINNT\PROFILES*username*\DESKTOP.
%*Favorites%	File system directory that serves as a common repository for the user's favorite items. Typically this directory is C:\WINDOWS\FAVORITES or C:\WINNT\PROFILES*username*\FAVORITES.
%*Fonts%	Virtual folder containing fonts. Typically C:\WINDOWS\FONTS or C:\WINNT\FONTS.
%*History%	File system directory that contains the user's history of visited Internet addresses. Typically C:\WINDOWS\HISTORY or C:\WINNT\PROFILES*username*\HISTORY.
%*NetHood%	File system directory containing objects that appear in the network neighborhood. Typically C:\WINDOWS\NETHOOD or C:\WINNT\PROFILES*username*\NETHOOD.
%*Personal%	File system directory that serves as a common repository for documents. Typically C:\MYFILES or C:\WINNT\PROFILES*username*\PERSONAL.
%*PrintHood%	File system directory that serves as a common repository for printer links. Typically C:\WINNT\PROFILES*username*\PRINTHOOD. Only used by Windows NT.

Macro	Description
%*Programs%	File system directory that contains the user's program groups (which are also file system directories). Typically C:\WINDOWS\START MENU\PROGRAMS or C:\WINNT\PROFILES*username*\START MENU\PROGRAMS.
%*Recent%	File system directory that contains the user's most recently used documents. Typically C:\WINDOWS\RECENT or C:\WINNT\PROFILES*username*\RECENT.
%*SendTo%	File system directory that contains Send To menu items. Typically C:\WINDOWS\SENTO or C:\WINNT\PROFILES*username*\SENDTO.
%*StartMenu%	File system directory containing Start menu items. Typically C:\WINDOWS\START MENU or C:\WINNT\PROFILES*username*\START MENU.
%*Startup%	File system directory that corresponds to the user's Startup program group. Typically C:\WINDOWS\START MENU\PROGRAMS\STARTUP or C:\WINNT\PROFILES*username*\START MENU\PROGRAMS\STARTUP.
%*TempDir%	Windows temporary directory (C:\WINDOWS\TEMP).
%*Templates%	File system directory that serves as a common repository for document templates. Typically C:\WINDOWS\SHELLNEW or C:\WINNT\SHELLNEW.
%*WinDesktop%	Windows desktop directory (C:\WINDOWS\DESKTOP or C:\WINNT\PROFILES*username*\DESKTOP for Windows NT). This is an Application Launcher 2.0 macro that is allowed for backward compatibility.
%*WinDir%	Windows directory. Typically C:\WINDOWS or C:\WINNT.
%*WinDisk%	Drive letter (plus colon) for the Windows directory. Typically C:.

Macro	Description
%*WinSysDir%	Windows system directory. Typically C:\WINDOWS\SYSTEM or C:\WINNT\SYSTEM32.
%*WinSysDisk%	Drive letter (plus colon) for the Windows system directory. Typically C:.

Windows 2000 Macros

The following macros apply to Windows 2000 only.

IMPORTANT: The asterisk character (*) is a required syntax for these macros. Don't confuse these asterisk characters with the Novell trademark asterisk. In addition, when using the macro in one of the Application object's property fields, you must enclose it in % characters (for example, %*WinDisk%).

Macro	Description
%*AdminTools%	File system directory that contains the administrative tools that appear in the Control Panel when a specific user logs in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS*username*\START MENU\PROGRAMS\ADMINISTRATIVE TOOLS.
%*CommonAdminTools%	File system directory that contains the administrative tools that appear in the Control Panel for all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\START MENU\PROGRAMS\ADMINISTRATIVE TOOLS.
%*CommonAppData%	File system directory that contains the application-specific data for all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\APPLICATION DATA.
%*CommonDocuments%	File system directory that contains the documents shared by all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\DOCUMENTS.

Macro	Description
%*CommonTemplates%	File system directory that contains the document templates shared by all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\TEMPLATES.
%*MyPictures%	File system directory that contains a specific user's graphic files. Typically C:\DOCUMENTS AND SETTINGS*username*\MY DOCUMENTS\MY PICTURES.
%*ProgramFiles%	File system directory that contains the program files. Typically C:\PROGRAM FILES.
%*ProgramFilesCommon%	File system directory that contains the program files shared by multiple applications. Typically C:\PROGRAM FILES\COMMON.
%*UserProfile%	File system directory that contains the logged in user's profile. Typically C:\DOCUMENTS AND SETTINGS*username*.

Login Script Macros

Application Launcher/Explorer supports many, but not all, login script variables.

- "Supported Login Script Macros" on page 225
- "Unsupported Login Script Macros" on page 227

For information about using the macros in a valid Application object property, see "Application Object Properties Where Macros Can Be Used" on page 231.

Supported Login Script Macros

The following is a list of supported login script variables and what they mean. Alternate variable names are shown in parentheses.

Macro	Description
%COMPUTER_NAME%	The name of the computer. For example: work_pc.

Macro	Description
%DAY%	Numeric day of the month. For example: 01, 10, 15.
%FILESERVER% (%FILE_SERVER%)	Name of the NetWare file server or NDS monitored connection. For example: APPS_PROD.
%FULL_NAME%	Full name attribute of the User object. This is supported by virtue of Application Launcher/ Explorer's support for NDS attributes from the User object.
%HOUR24% (%24HOUR%)	Time of the day according to a 24-hour clock. For example: 02, 05, 14, 22.
%HOUR% (%HOURS%)	Hour of the day. For example: 0 = 12, 13 = 1.
%LAST_NAME%	Last name of the current user (also known as the user's NDS Surname attribute). For example: Jones.
%LOGIN_NAME%	First eight bytes of the user's NDS object name. For example: jsmith.
%MINUTE% (%MINUTES%)	Current minute. For example: 02, 59.
%MONTH%	Current month number. For example: 01 for January.
%NDAY_OF_WEEK%	Numeric day of the week. For example: 1 for Sunday, 2 for Monday.
%NETWORK% (%NETWORK_ADDRESS%)	Workstation network address. For example: 01010120.
%OS_VERSION%	Version of the OS. For example: v5.00 (Win3*x* shows the DOS version, Win 95/98 and Win NT/ 2000 shows the Windows version).
%OS	OS type. For example: MSDOS, WIN95/98, WINNT, WIN2000. (Win3 shows MSDOS.)
%PLATFORM%	Platform running. For example: WIN, W95/98, WNT, W2000.

Macro	Description
%PHYSICAL_STATION% (%P_STATION%)	MAC address. For example: 0000C04FD92ECA.
%REQUESTER_CONTEXT%	Context of the requester for the selected tree.
%SECOND% (%SECONDS%)	Number of seconds. For example: 03, 54.
%SHORT_YEAR%	Short year number. For example: 97, 00.
%WINVER%	Windows version. For example: v3.11, v4.00.
%WORKSTATION_ID%	The name of the workstation object.
%WORKSTATION_TREE%	The name of the tree in which the workstation is located.
%WORKSTATION_DN%	The distinguished name and context of the workstation object.
%YEAR%	Full year number. For example: 2000, 2001.

Unsupported Login Script Macros

The following is a list of login script macros that Application Launcher/ Explorer does not support:

ACCESS_SERVER
AM_PM
DAY_OF_WEEK
DIALUP
ERROR_LEVEL
GREETING_TIME
LOCATION
LOGIN_ALIAS_CONTEXT
LOGIN_CONTEXT
MACHINE
MONTH_NAME
NEW_MAIL
OFFLINE

PASSWORD_EXPIRES
REQUESTER_VERSION
SHELL_VERSION
SMACHINE
STATION
USERID

NDS Attribute Macros

Application Launcher/Explorer supports macros that pull information from the attributes of the currently logged-in user, the current Application object, or from the attributes of other NDS objects.

The following sections explain the macro syntax and provide examples:

- “Syntax” on page 228
- “Examples” on page 229

For information about using the macros in a valid Application object property, see “Application Object Properties Where Macros Can Be Used” on page 231.

Syntax

NDS attribute macros use the following syntax:

```
%object_name.container;NDS_attribute%
```

Element	Description
%	Flags the text as a macro. The entire macro must be enclosed in % characters.

Element	Description
.object_name.container;	Indicates the NDS object from which to read the attribute. To use an attribute from the currently logged in user, leave this element out. For example, %CN% would return the common name of the currently logged in user. To use an attribute from the Application object, replace *.object_name.container* with an asterisk (*). For example, %*;DN% would return the distinguished name of the Application object.
NDS_attribute	Defines the attribute to be read. You can use the ConsoleOne Schema Manager (available from the Tools menu) to view an NDS object's available attributes.

Examples

The following table provides examples of NDS attribute macros.

Macro	Description
%CN%	Returns the common name of the currently logged-in user.
%DN%	Returns the distinguished name of the currently logged-in user.
%Given Name%	Returns the first name of the currently logged-in user.
%Surname%	Returns the last name of the currently logged-in user.
%Full Name%	Returns the full name of the currently logged-in user.

Macro	Description
%Email Address%	Returns the e-mail address of the currently logged-in user.
%*;DN%	Returns the distinguished name of the Application object (the one in which the macro is being used).
%*;App:Caption%	Returns the application icon title given the Application object.
*;App:Path%	Returns the path to the application's executable as defined for the Application object.
%*;App:GUID%	Returns the Application object's global unique identifier (GUID).
%.JSmith.Novell;Description%	Returns the description for the JSmtih User object located in the Novell container.
%.ABurrough.Novell;Email Address%	Returns the e-mail address for the ABurrough User object located in the Novell container.

Environment Variable Macros

Application Launcher/Explorer supports all valid Windows 95/98 and Windows NT/2000 environment variables. The value of the variable must not exceed the length of the Application object name; otherwise, the variable fails.

Here are a few examples:

- NWLANGUAGE
- TEMP
- PATH

When using an environment variable, you must enclose it in % characters (%PATH%).

For information about using the macros in a valid Application object property, see "Application Object Properties Where Macros Can Be Used" on page 231.

Macro Precedence

To resolve conflicts with macro names, Application Launcher/Explorer uses the following order of precedence:

- Application object macros
- Special Windows macros
- Login script macros
- NDS attribute macros
- Environment variable macros.

The macro types are listed in order of precedence. For example, Application object macros take precedence over special Windows macros, special Windows macros take precedence over login script macros, and so forth.

Application Object Properties Where Macros Can Be Used

You can use macros when defining the following Application object properties:

- Path to Executable (Run Options tab > Applications page)
- Command Line (Run Options tab > Applications page)
- Working Directory (Run Options tab > Applications page)
- Drive Mapping Path (Common tab > Drives/Ports page)
- Capture Port Path (Common tab > Drives/Ports page)
- Key, Value Name, and Value Data strings (Distribution Options tab > Registry Settings page)
- Section, Value Name, and Value Data strings (Distribution Options tab > INI Settings page)
- Source File, Target File, Directory Name (Distribution Options tab > Application Files page)
- Find File, Find Next File, Add string (Distribution Options tab > Text Files page)
- All properties (Distribution Options tab > Icons/Shortcuts page)

When entering a macro in a property field, you must enclose the macros in % characters (for example, %TARGET_PATH% or %*WINDISK%). This enables Application Launcher/Explorer to recognize and treat the macro as a macro.

14 Application Launcher Tools

ZENworks™ for Desktops (ZfD) provides several tools that may be helpful as you manage applications for your users. These tools, located on the Tools > Application Launcher Tools menu in ConsoleOne™, are explained in the following sections:

- "Export Application Object" on page 233
- "Show Inherited Applications" on page 233
- "Search and Replace" on page 234
- "Sync Distribution GUIDs" on page 235
- "Generate New GUIDs" on page 235
- "AOT/AXT File Tools" on page 235
- "Create Virtual CD" on page 236

Export Application Object

This tool enables you to export an Application object's properties to a new .AOT or .AXT file. You can perform a default export, which exports all general properties, INI settings, registry settings, text file changes, shortcuts, environment variables, and application files, or you can perform a custom export, which enables you to choose which settings you want exported.

1. In ConsoleOne, select the Application object.
2. Click the Tools menu > click Application Launcher Tools > click Export Application Object.
3. Select the export option you want > click Next > follow the prompts to complete the export.

Show Inherited Applications

This tool enables you to see the applications that have been associated with a user or workstation, including applications associated with groups or containers that the user or workstation belong to. The applications are listed by mode of delivery, such as Force Run, App Launcher, Desktop, Start Menu, and System Tray.

1. In ConsoleOne, select a User object or Workstation object.
2. Click the Tools menu > click Application Launcher Tools > click Show Inherited Applications.

Search and Replace

This tool lets you search and replace text strings in an Application object's general settings, registry settings, INI settings, copy files settings, Icon settings, and text file settings. As settings are found that match your search criteria, you are given the choice of whether or not to replace the text string.

1. In ConsoleOne, select the Application object.
2. Click the Tools menu > click Application Launcher Tools > click Search and Replace.
3. In the Search For box, type the text string you are searching for.
4. In the Replace With box, type the text string you want to use in place of the text string you are searching for.
5. Check the Match Case box to require the search results to match the case (uppercase and lowercase) of the text string you entered in the Search For box.
6. If you want to limit the Application object settings that are searched, click Options > deselect any settings you don't want searched > click OK to return to the Search and Replace dialog box.
7. To start the search, click Find Next.

   The dialog box expands to display the search results. If a setting is found that matches the search criteria, the setting's text is displayed in the Text Found box.
8. To replace the setting's text with the text listed in the Replace With box, click Replace.

   or

   To replace the text for all settings that match the criteria, click Replace All.

   or

   To skip replacing the setting's text, click Find Next.
9. Repeat Step 8 until all settings have been searched.

Sync Distribution GUIDs

This tool enables you to synchronize the GUIDs (global unique identifiers) of two or more Application objects. Synchronizing GUIDs causes all Application objects to be distributed and installed whenever one Application object is. This can be useful, for example, when you want to distribute a suite of applications to users.

1. In ConsoleOne, select the Application objects whose GUIDs you want to synchronize. Control-click to select multiple objects.
2. Click the Tools menu > click Application Launcher Tools > click Sync Distribution GUIDs.

Generate New GUIDs

This tool generates a new GUID for an Application object. If you've synchronized the GUIDs of several Application objects, you can use this tool unsynchronize the GUIDs so that distributing one of the Application objects will not distribute all of them.

This tool requires careful use. When an Application object is distributed to a workstation, its GUID is stored in the Windows registry. Novell® Application Launcher™/Explorer uses the GUID to determine whether or not the application has been distributed to the workstation. After the new GUID is generated, Application Launcher/Explorer may re-distribute the application because the GUID in the Windows registry will not match the new GUID for Application object.

1. In ConsoleOne, select the Application objects whose GUIDs you want to change. Control-click to select multiple objects.
2. Click the Tools menu > click Application Launcher Tools > click Generate New GUIDs.

AOT/AXT File Tools

This tool converts a .AOT file to a .AXT file or a .AXT file to a .AOT file.

1. In ConsoleOne, click the Tools menu > click Application Launcher Tools > click AOT/AXT File Tools.
2. To convert a .AOT file to a .AXT file, click AOT -> AXT.

or

To convert a .AXT file to a .AOT file, click AXT -> AOT.

3. In the Open dialog box, browse for and select the file you want to convert > click Open.
4. In the Save As dialog box, type a name for the new file > click Save As. The correct extension (.AOT or .AXT) will be added automatically.

Create Virtual CD

This tool lets you create a cache of applications that can be added to removable media (CD, JAZ* drive, ZIP* drive, and so forth). The media can then be given to users who are disconnected from the network so that they can install the applications from the CD. For detailed information and instructions, see "Distributing Applications to Disconnected Workstations" on page 84.

IV Workstation Imaging

The following sections provide information on ZENworks™ for Desktops (ZfD) Workstation Imaging procedures and features that you might need to use generally or from time to time as required.

- Chapter 15, “Preparing Images,” on page 245
- Chapter 16, “Multicasting Images,” on page 251
- Chapter 17, “Imaging Utilities and Options,” on page 255
- Chapter 18, “Supported Ethernet Cards,” on page 287

15 Preparing Images

ZENworks™ for Desktops (ZfD) provides tools for creating images of computer hard disks as well as images of specific add-on applications or file sets. ZfD also provides tools for customizing such images and for making images available to auto-imaging operations through NDS. The following sections explain how to perform these tasks.

- “Creating a Workstation (Base) Image” on page 245
- “Creating an Add-On Image” on page 246
- “Customizing an Image” on page 247
- “Making an Image Available for Auto-Imaging” on page 247

Creating a Workstation (Base) Image

A *base* image is an image of all the partitions and data on a source computer’s storage devices, such as hard disks and Jaz drives. Normally, such an image is prepared with the intent to completely replace the contents of some other target computer’s storage devices.

The overall process to create a base image is:

1. Boot the source computer from an imaging device.

   The boot device can be diskettes, a CD, or a hard-disk partition.

2. Run the ZfD imaging engine to take an image of the computer.

   You can do this manually or automatically. In manual mode, you interrupt the bootup process and issue a certain imaging command at the Linux prompt. In automatic mode, you set a flag in the computer’s Workstation object in NDS and then let the bootup process proceed without interruption.

The instructions you should follow to complete this process depend on the boot device and imaging mode you want to use, as summarized in the table below.

Boot device / Imaging mode	For instructions, see
diskettes / manual	Taking an Image of a Workstation Manually in Testing Basic Imaging Operations in Workstation Imaging in *Getting Started*
CD or partition / manual	Preparing an Imaging Boot Device in Setting Up Workstations for Imaging in Workstation Imaging in *Deployment*
any device / automatic	Triggering an Unattended Imaging Operation in Testing Basic Imaging Operations in Workstation Imaging in *Getting Started*

Creating an Add-On Image

An *add-on* image is an archived collection of files that will be applied to an existing Windows installation on a target computer. The existing partitions and files on the target computer are left intact, except for any files that the add-on image might update.

An add-on image typically corresponds to an application or utility, or simply to a set of data files or configuration settings. There are two ways to create an add-on image, each resulting in a different kind of add-on image:

- Produce the add-on image from an Application object

  You do this in ConsoleOne by using the Imaging property page (on the Common tab) of the Application object. For details, click Help on that property page.

  An add-on image created in this manner is not fully installed on the target computer until after the computer reboots from being imaged and Application Launcher/Explorer starts up and runs the new Application object. Deploying the image is essentially another way to distribute an Application object.

- Drag files into a new image archive using Image Explorer

  You do this by starting Image Explorer, dragging files and folders from an existing Windows installation into the new image archive, and saving the archive to a file with a .ZMG extension. For more information, see “Image Explorer (IMGEXP.EXE)” on page 255.

  Generally, an add-on image created in this manner doesn’t require any post processing on the target computer. It is simply a set of files that are copied to the appropriate locations on the hard disk, much like what happens when you unzip a WinZip archive. One exception is that the add-

on image can contain Windows registry (.REG) files that will be applied to the registry automatically when the computer reboots after being imaged, if the imaging agent is installed on the computer. For more information, see the next section.

Customizing an Image

Once you have created a base or add-on image as explained in the previous sections, you can customize it using the Image Explorer utility. Specifically, you can:

- Exclude individual files and folders from the image

  In doing this, you create variants of the image by specifying which of ten possible *filesets* (variants) to exclude a given file or folder from. The variants exist merely as internal attributes of the same image archive.

  WARNING: Do not exclude .BPB files from a base image or the workstation won't be able to boot the new operating system after receiving the image.

- Add files and folders to the image

  By default, any file or folder you add is included in all variants. To change this, you must explicitly exclude the file or folder from one or more variants.

- Add Windows registry (.REG) files

  The registry settings contained in the .REG files that you add are applied after the image is laid down and the target computer reboots to Windows, if the imaging agent has been installed on the computer.

  As with any other file or folder that you add, a .REG file is included in all variants of the image unless you explicitly exclude it from one or more variants.

For information on starting Image Explorer, see "Image Explorer (IMGEXP.EXE)" on page 255. For information on how to complete the above tasks once you have started the utility, see the online help in the utility.

Making an Image Available for Auto-Imaging

When you boot a computer from an imaging device and allow the bootup process to proceed in auto-imaging mode, the imaging operation that is

performed on the computer is determined by policies and settings that you define in NDS.

In order to make an image available to such operations, you must expose it as a Workstation Image object in NDS. Otherwise, when you define imaging policies and settings in NDS, you won't have any way to refer to the image. Creating a Workstation Image object also allows you to combine a base image and one or more add-on images into a single entity that can be laid down on target computers. You can also specify that a particular variant of an image be used. The sections that follow give instructions for performing these tasks.

- "Creating a Workstation Image Object" on page 248
- "Associating an Add-On Image with a Base Image" on page 249
- "Specifying to Use a Variant of an Image" on page 249

Creating a Workstation Image Object

1. Create the base image that the Workstation Image object will refer to. For instructions, see "Creating a Workstation (Base) Image" on page 245.

   HINT: Although it isn't typical, you can create a Workstation Image object that refers only to one or more add-on images. However, if you want a base image to be laid down in the same operation as the add-on images, you should include both types of images in the Workstation Image object.

2. Copy the image file to a ZfD imaging server that is accessible as a server object in your NDS tree.
3. In ConsoleOne, open the NDS tree and browse to the container where you want to create the Workstation Image object.
4. Right-click the container > click New > Object > select Workstation Image from the list of object classes > click OK.
5. Enter a name for the new object.

   Example: `Dell NT4 Image`
6. Click Define Additional Properties > OK.
7. Under Base Image File, click the browse button next to the field > select the imaging server where the image resides > select or specify the path and filename of the image > click OK.

   For details on selecting or specifying the path and filename, click Help in the Image File Location dialog box.

HINT: If the Workstation Image object will consist only of add-on images, leave the Base Image File field blank, and skip to Step 5 of "Associating an Add-On Image with a Base Image" on page 249.

8 Click OK to save the Workstation Imaging object.

Associating an Add-On Image with a Base Image

1 Create the add-on image that you will associate with the base image. For instructions, see "Creating an Add-On Image" on page 246.

2 Copy the image file to a ZfD imaging server that is accessible as a server object in your NDS tree.

You might want to copy the add-on image to the same location as the base image.

3 In ConsoleOne, open the NDS tree and browse to the Workstation Image object that refers to the base image. If you haven't created this object yet, do so as instructed in "Creating a Workstation Image Object" on page 248.

4 Right-click the object > click Properties.

5 Under Add-on Image Files, click the Add button > select the imaging server where the add-on image resides > select or specify the path and filename of the image > click OK.

For details on selecting or specifying the path and filename, click Help in the Image File Location dialog box.

HINT: You can associate more than one add-on image with a base image. The add-on images will be laid down after the base image in the order listed on this page.

6 Click OK to save the Workstation Imaging object.

Specifying to Use a Variant of an Image

As explained in "Customizing an Image" on page 247, you can exclude individual files and folders from any of 10 possible *filesets* (variants) of an image. The variants exist merely as internal attributes of the same image archive.

Because creating an image of a workstation can take a fair amount of time, it is more efficient in some cases to just create an image of a few workstations and customize those images to get all the variants you need. Even though the

variants do not all exist as separate, physical image files, you can access them as though they were. How you do this depends on whether you are performing a manual or automated imaging operation, as explained below.

Type of imaging operation	How to specify the variant to use
Automatic (NDS-based)	In the Workstation Image object, specify the number of the variant in the Use File Set field. All NDS policies and settings that specify that Workstation Image object will use the specified variant. **HINT:** You can create multiple Workstation Image objects that point to the same base image but to different variants.
Manual (command-line)	Use the **s** parameter on the **img r** command. For example, to specify variant number 3: **img rl dellnt4.zmg s3** For details, see "Imaging Engine (img Command)" on page 264.

16 Multicasting Images

ZENworks™ for Desktops (ZfD) Imaging software includes an imaging multicast capability. The following sections explain what this is, why you might want to use it, and the overall procedure for using it.

- "What Is Multicasting?" on page 251
- "Why Use Multicast?" on page 251
- "Multicast Procedure" on page 252

What Is Multicasting?

To *multicast* an image is to take an image of one computer (the *master*), immediately send it over the network to multiple other computers (the *slaves*), and lay it down simultaneously on those computers. No NDS or ZfD server involvement is required.

As with all imaging operations involving a base image, the image is taken of all the partitions on the hard disks and other storage devices (such as Jaz drives) of the master computer. Before the image is laid down on the slave computers, all existing partitions are removed from the hard disks and writeable storage devices of those computers.

To conduct a multicast session, each computer must be physically visited to initiate participation in the session. You cannot set up automatic (unattended) multicast sessions in this release of ZfD.

NOTE: For multicasting to work properly, the routers and switches on the network must have multicast features configured. Otherwise, multicast packets might not be routed properly.

Why Use Multicast?

Multicasting is the way to use ZfD imaging services for mass reimaging with the least amount of overhead. It is useful if you have one computer with a clean software configuration that you want to clone on several other machines without having to prepare an image or CD, and without having to install or configure any ZfD software on a server or on the target computers.

With multicasting, all you need is a physical network with modern routers and switches and two ZfD imaging boot diskettes, which are very easy to prepare. The computers that will be imaged must be physically connected to the network. They can be computers with existing operating systems of any kind, or they can be brand new computers with no operating system installed.

Limitations

One significant limitation of using multicast without installing any ZfD software is that it results in a set of computers that have duplicate network identities—the IP addresses, Computer (NETBIOS) names, Workgroup memberships, and Security Identifiers (Windows NT/2000 only) are all the same and thus will cause conflicts if deployed on the network without change.

For a handful of computers, this might not be a problem. But for a larger number, if the computers have Windows, we recommend that you install the ZfD imaging agent on them before doing the multicast. (You can use an Application object to do this easily. See Installing the Imaging Agent to Safeguard Workstation Identity Data in Setting Up Workstations for Imaging in Workstation Imaging in *Deployment*.) The imaging agent saves off the computer's network identity settings before the multicast session and restores them afterwards.

Multicast Procedure

There are a couple of ways to conduct a multicast session. You can:

- Physically visit each computer yourself and start the multicast session on those computers one at a time
- Enlist several people to attend the various computers so that the multicast session can be started more or less simultaneously on all the computers.

Either way, here's the overall procedure:

1. (Optional) Install the ZfD imaging agent on each of the target computers.

   See "Why Use Multicast?" on page 251 for the reasons and for further instructions.

2. Create a set of ZfD imaging boot diskettes for each person who will assist with the multicast session.

   If you don't know how to do this yet, see Creating Imaging Boot Diskettes in Preparing for Basic Imaging Operations in Workstation Imaging in *Getting Started*.

3 At each computer, including the master computer that you will clone, do the following to access a Linux prompt:

3a Insert the first imaging diskette and reboot the computer.

3b As soon as the boot prompt appears, type **`manual`** > press Enter.

IMPORTANT: Don't leave the computer unattended after booting it, or it will go into automatic mode and you won't be able to access the Linux prompt.

3c When prompted for the second diskette, insert it > press Enter.

4 At the Linux prompt of each computer, enter the following command to start the multicast session:

`img s` *`session_name`*

where *session_name* is any string that uniquely identifies this multicast session from other multicast sessions that might be in progress on the network. Use the same session name on each of the computers in this multicast session.

Example: **`img s doug`**

5 At the master computer, type **`m`** > press Enter.

6 At the master computer, after all the other computers have registered as slaves, type **`g`** > press Enter.

The imaging engine begins creating the image of the master computer. Once the image is done, it is sent to and laid down on each slave computer. Any problems are reported back and displayed on the master computer.

7 At each slave computer, when the imaging is done, do the following to boot the computer with the new operating system:

7a At the Linux prompt, type **`export IMGSUCCESS=1`** > press Enter.

IMPORTANT: This command is case-sensitive.

7b At the Linux prompt, type **`lilo.s`** > press Enter.

7c Press Ctrl+Alt+Delete.

HINT: If the computer doesn't boot to the new operating system (that is, if the Linux prompt reappears), enter the **`lilo.s`** command again and reboot the computer a second time.

The **`img s`** command can take other parameters that allow you to designate the master computer and the imaging start time beforehand. See "Imaging Engine (img Command)" on page 264 for details.

17 Imaging Utilities and Options

The following sections provide reference information on ZENworks™ for Desktops (ZfD) Imaging utilities, commands, configuration settings, and log formats.

- "Image Explorer (IMGEXP.EXE)" on page 255
- "Imaging Agent (ZISWIN.EXE)" on page 256
- "Imaging Boot Disk Creator (ZIMGBOOT.EXE)" on page 258
- "Imaging Bootup Parameters (SETTINGS.TXT)" on page 260
- "Imaging Bootup Languages (ZIMGLANG.INI)" on page 264
- "Imaging Engine (img Command)" on page 264
- "Imaging Server (IMGSERV.NLM or .DLL or .DLM)" on page 281
- "Imaging Server Log (ZIMGLOG.XML)" on page 283

Image Explorer (IMGEXP.EXE)

Purpose

Use this utility at a Windows workstation to view or customize workstation images or to create add-on images.

Location

This utility is located in the ZENWORKS\IMAGING folder in your ZfD installation (on the imaging server).

Format and Function

To start this utility stand-alone (from Windows), double-click the IMGEXP.EXE file. There are no command-line parameters. To start the utility from ConsoleOne, click Tools > ZENworks Utilities > Imaging > Image Explorer.

After starting the utility, you can create a new add-on image or open an existing workstation image. You can browse and view the Windows partitions, folders, and files in the open image. You can customize the image by adding or excluding individual files and folders and by adding Windows

registry (.REG) files that will be applied after the image is laid down. You can associate these customizations with any of ten possible variants of the image. For information on how to perform these tasks, see the online help in the utility.

WARNING: Do not exclude .BPB files from a base image or the workstation won't be able to boot the new operating system after receiving the image.

NOTE: Non-Windows partitions, such as NetWare partitions, are visible when you open an image, but their contents are not.

Imaging Agent (ZISWIN.EXE)

Purpose

The imaging agent is an extension to the Windows bootup procedure on a workstation. It runs before any network communications are established. It enables you to:

- Make an existing Windows workstation safe for reimaging

  When you install the imaging agent on an existing Windows workstation, it saves off certain workstation-unique data (such as the IP address and Computer name) to an area on the hard disk that is safe from reimaging. After the workstation is reimaged and Windows reboots, the agent restores the data from the image-safe area so the workstation can communicate on the network using the same identity as before.

- Assign a network identity to a brand new workstation automatically

  A brand new workstation (with no Windows operating system) doesn't have a network identity established yet. You can define network identity information for such a workstation in an NDS policy and apply it when the workstation receives its first Windows image. In this scenario, the Windows image is laid down (including the imaging agent) on the workstation and the identity information from the NDS policy is written to the image-safe area on the hard disk. When the workstation reboots, the imaging agent reads the data from the image-safe area and applies it to the Windows installation, thus establishing the workstation's network identity automatically.

Location

Before you install the imaging agent on a workstation, the files that comprise it are available in the ZENWORKS\IMAGING folder in your ZfD installation

(on the imaging server). After you install the imaging agent on a workstation, it is located either in the Windows system folder or in NOVELL\ZENIS on the drive where Windows is installed.

Format and Function

To install the imaging agent so that it runs automatically each time Windows starts, follow the instructions in Setting Up Workstations for Imaging in Workstation Imaging in *Deployment*.

The data that the imaging agent saves to (or restores from) the image-safe area includes the following:

- Whether a static IP address or DHCP is used
- If a static IP address is used:
  - IP address
  - Subnet mask
  - Default gateway (router)
- Computer (NETBIOS) name
- Workgroup that the workstation belongs to, if any
- If the workstation has been registered in NDS:
  - Distinguished name of the Workstation object
  - Context of the Workstation object
  - NDS tree that the Workstation object belongs to

On a workstation that has just received a new Windows NT/2000 base image, in addition to restoring the above date, the agent also locates and modifies all instances of the Security Identifier (SID). This ensures that the workstation has a SID that is unique from other workstations that might receive the same image.

IMPORTANT: The imaging agent does not save off or restore any Windows NT/2000 Domain information. If you change a workstation's domain and then restore an image, the workstation will receive whatever domain is embedded in the new image.

Imaging Boot Disk Creator (ZIMGBOOT.EXE)

Purpose

Use this utility at a Windows workstation to create or update imaging boot diskettes, so you can boot computers to perform imaging tasks.

IMPORTANT: You can update imaging diskettes only in the sense that the same physical diskettes can be reused. When updating the second diskette or the optional language diskette, you must reformat the diskette before updating it. The first diskette can be reused without reformatting.

Location

This utility is located in the ZENWORKS\IMAGING folder in your ZfD installation (on the imaging server).

Format and Function

To start this utility stand-alone (from Windows), double-click the ZIMGBOOT.EXE file. There are no command-line parameters. To start the utility from ConsoleOne, click Tools > ZENworks Utilities > Imaging > Create or Modify Boot Diskette.

For the overall process to create imaging boot diskettes, see Creating Imaging Boot Diskettes in Workstation Imaging in *Getting Started*. When creating the diskettes, you can configure various aspects of the imaging bootup process, including:

- How to communicate on the IP network, if necessary
- How automated the imaging bootup process should be
- Which imaging server to contact during automated operations
- How large the imaging (Linux) partition on the hard disk should be, if one will be created
- Which language support (English or otherwise) to load for the keyboard

For details on these configuration options, see the online help in the utility. The configuration settings you make in the utility are saved to the SETTINGS.TXT file on the second imaging diskette. You can edit this file after creating the diskettes if needed. For details on the format of this file, see "Imaging Bootup Parameters (SETTINGS.TXT)" on page 260.

If you need to create imaging diskettes to boot with a non-English keyboard but the language you need isn't listed in the utility, see "Imaging Bootup Languages (ZIMGLANG.INI)" on page 264.

Using ZIMGBOOT.EXE to Add Linux Drivers

NOTE: Support Pack 1

All of the information in this section pertains only to the version of ZIMGBOOT.EXE shipping in ZfD3 Support Pack 1 (ZfD3 SP1).

The ZfD3 SP1 version of the Imaging Boot Disk Creator (ZIMGBOOT.EXE) has an Add Linux Drivers function where you can specify the Linux* drivers you want to place on the second boot diskette (if space allows) or an "additional Linux drivers" diskette.

Using this function, you can search for and add the network path to Linux driver files that you have created or downloaded. The dialog box helps you to build a drivers list and categorize it by driver type (SCSI, Block, Network, PCMCIA, and Miscellaneous). You can also remove unwanted drivers from the list. This master list of driver files can be added to a diskette that you can specify later.

The Add Linux Drivers function also lets you specify the drivers you want to load by default. You do this by selecting the name of a driver on the master list and then clicking the Load button. This moves the driver name to the default load list, where you can change the load order of the default drivers and further specify their individual loading parameters.

When your load list is ready, you can use a function in ZIMGBOOT.EXE to copy the additional Linux drivers to a diskette. All of the driver files are stored in different subdirectories of the \DRIVERS directory on the diskette:

- Network drivers are stored in \DRIVERS\NET
- PCMCIA drivers are stored in \DRIVERS\PCMCIA
- Block drivers are stored in \DRIVERS\BLOCK
- SCSI drivers are stored in \DRIVERS\SCSI
- Miscellaneous drivers are stored in \DRIVERS\MISC

These drivers can also be added to a bootable CD. For more information, see Preparing a Bootable CD in Preparing an Imaging Boot Device in Setting Up Workstations for Imaging in *Deployment*. The information is unique to ZfD3 SP1.

Obtaining Linux Drivers

NOTE: Support Pack 1

All of the information in this section pertains only to the version of ZIMGBOOT.EXE shipping in ZfD3 Support Pack 1 (ZfD3 SP1).

To obtain a Linux driver for your particular hardware, you should visit the Web site of the hardware vendor and check for a download site.

There are also some other Web sites where you can obtain drivers:

- Network drivers can be downloaded from the Scyld Computing Corporation* at www.scyld.com/linux_network_drivers.html (http://www.scyld.com/linux_network_drivers.html).
- PCMCIA drivers can be downloaded from the Linux PCMCIA Information Page at pcmcia-cs.scourceforge.net (http://pcmcia-cs.sourceforge.net/).

You can also get additional Linux drivers at the Novell® ZENworks Cool Solutions Web Community at www.novell.com/coolsolutions/zenworks/features/a_linux_drivers_zw.html (http://www.novell.com/coolsolutions/zenworks/features/a_linux_drivers_zw.html).

To learn more about drivers, for example, the loading parameters you need to specify, see the Linux Documentation Project at www.linuxdoc.org (http://www.linuxdoc.org/) and visit the following HOWTO (http://www.linuxdoc.org/HOWTO/HOWTO-INDEX/howtos.html) sites:

- Hardware
- PCMCIA
- SCSI
- Ethernet

Imaging Bootup Parameters (SETTINGS.TXT)

Purpose

This file contains parameters that control how the imaging boot-up process occurs.

Location

This file is installed to the root of the imaging boot device (CD, hard-disk partition, or the second diskette).

Format and Function

This is a plain text file that contains various parameters, each on a separate line. Each parameter has the general format of `PARAMETER=value`. Lines that begin with a pound sign (#) are comments and are ignored during the imaging bootup process.

The format and function of each parameter in the SETTINGS.TXT file are described in the table below.

Parameter	Specifies
PROMPT	Whether to prompt for each configuration setting when you boot a computer from the imaging device. If you leave this parameter commented out or set it to No, the computer boots using the configuration settings specified in SETTING.TXT and you can't override the settings during bootup unless you type **`config`** at the boot prompt before the Linux operating system begins to load. If you set this parameter to Yes, you are automatically prompted for each configuration setting during bootup.
MANUALREBOOT	Whether you must reboot a computer manually after it was booted from the imaging device in automatic mode. (If the computer was booted from the imaging device in manual mode, you must always reboot the computer manually.) If you boot a computer from the imaging device and you let the bootup process proceed in automatic mode, the imaging engine starts up and checks the imaging server to see if an imaging operation should be performed on the computer. If so, it performs the imaging operation and then quits. If not, it quits without doing anything. What happens next depends on how you set this parameter. If you leave it commented out or set it to No, you are prompted to remove the imaging device (if necessary) and press any key to reboot the computer automatically to the native operating system. If you set this parameter to Yes, the computer doesn't reboot automatically but instead displays the Linux prompt, allowing you to perform additional imaging-related tasks at the command line. This is helpful if you want to do things like check the current partition information or the image-safe data before booting to the native operating system.

Parameter	Specifies
PARTITIONSIZE	The number of megabtyes to allocate to the imaging (Linux) partition if you choose to create one locally on a computer when you boot the computer from the imaging device. The default size is 15 MB, which is the smallest you should make the partition. The maximum size allowed is 2048 MB (2 GB). If you plan to store an image in the Linux partition, such as to enable the computer to be restored to a certain state without connecting to the network, you might want to specify a larger size on this parameter. Example: `PARTITIONSIZE=500`
IPADDR	The IP address used by a computer to communicate on the network when you boot the computer from the imaging device, if a static IP address is needed. Example: `IPADDR=137.65.95.126` If you want DHCP to be used, leave this and the next two parameters commented out.
NETMASK	The subnet mask to be used by the computer, if the computer is using a static IP address. Example: `NETMASK=255.255.252.0` If DHCP is being used, leave this parameter commented out.
GATEWAY	The IP address of the gateway (router) to be used by the computer, if the computer is using a static IP address. Example: `GATEWAY=137.65.95.254` If DHCP is being used, leave this parameter commented out.

PROXYADDR	The IP address or full DNS name of the imaging (proxy) server to connect to when you boot a computer from the imaging device in auto-imaging mode.
	
	Examples:
	`PROXYADDR=137.65.95.127`
	`PROXYADDR=imaging.xyz.com`
	
	This parameter is used to set the PROXYADDR environment variable in Linux when the computer is booted from the imaging device. The imaging engine then reads this variable to determine which server to contact if it's running in automatic mode. Whether it's running in automatic or manual mode, the imaging engine attempts to log the imaging results to the server specified in this variable.

Parameter	Specifies
DEFAULTDRIVER	The default Linux Ethernet driver used by a computer to communicate on the network when you boot the computer from the imaging device. In most cases you can leave this blank and the driver will be autodetected. However, if the computer happens not to have one of the "Supported Ethernet Cards" on page 287, you must get a Linux driver for it, put the driver on the imaging boot device, and specify the driver filename on this parameter. Example: `DEFAULTDRIVER=3C05X.O` You can get Linux drivers from various places on the Web, such as the Linux Network Drivers (http://www.scyld.com/network/) Web site. If the driver you get is a PCMCIA driver (for a laptop computer), you must create a drivers/pcmcia directory on the imaging boot device (CD or the second diskette) and put the driver there. If the driver you get is non-PCMCIA, you must create a drivers/net directory and put the driver there. **NOTE: Support Pack 1** New code introduced with Support Pack 1 now makes this setting obsolete. Access the Help in ZIMGBOOT.EXE for information about the Load Additional Drivers setting or see "Using ZIMGBOOT.EXE to Add Linux Drivers" on page 259. Driver information will not be saved in SETTINGS.TXT.
DEFAULTPORT	The port (interrupt) number used by a computer to communicate on the network when you boot the computer from the imaging device. In most cases you can leave this blank and the port will be autodetected. However, with some older Ethernet cards, you might need to specify the port to get the card to work. (See the documentation for the Ethernet card for details.) Enter the port number in hexadecimal (digits 0-9, letters A-F). Example: `DEFAULTPORT=300` **NOTE: Support Pack 1** New code introduced in Support Pack 1 ZIMGBOOT.EXE now makes this setting obsolete. The information is not saved in SETTINGS.TXT. Access the online Help in ZIMGBOOT.EXE or see "Using ZIMGBOOT.EXE to Add Linux Drivers" on page 259 for more information.
LANGDISK	Whether to prompt for a language diskette when you boot a computer from the imaging device. Set this parameter to Yes only if the computer has a non-English keyboard and you have created a language diskette to support that keyboard as explained in the online help for the Imaging Boot Disk Creator (ZIMGBOOT.EXE) utility. If you need to support a language that's not listed in the Imaging Boot Disk Creator utility, see "Imaging Bootup Languages (ZIMGLANG.INI)" on page 264.

Imaging Bootup Languages (ZIMGLANG.INI)

Purpose

This file defines the non-English keyboards that are supported for the imaging bootup process. You can create language diskettes for each of these languages using the Imaging Boot Disk Creator (ZIMGBOOT.EXE) utility. You can add support for additional languages by extending this file.

Location

This file is located in the ZENWORKS\IMAGING folder in your ZfD installation (on the imaging server).

Format and Function

This is a standard Windows INI format file. Each section of the file defines the keyboard support for a single language, including the keyboard mappings, fonts, and Unicode mappings to use. For example, the German keyboard is defined like this:

```
[German]
keymap=keymaps/de.kmap.gz
Font=consolefonts/iso01.f16.psf.gz
ACM=consoletrans/iso01.acm.gz
```

You can add support for additional languages to this file as explained in Setting Up Workstations for Imaging in Workstation Imaging in *Deployment*.

Imaging Engine (img Command)

Purpose

Use this command at the Linux prompt after booting a computer from an imaging device to do any of the following:

- Take an image of the computer's hard disks
- Lay down an image on the computer's hard disks
- View or manipulate the computer's hard disk partitions
- View the computer's hardware configuration or image-safe data

Location

The imaging engine is installed to the bin folder on the imaging boot device. If the imaging boot device is dikettes or a CD, the bin folder is actually archived in the root.tgz file, which gets expanded during the imaging bootup process.

Format and Function

Because the imaging engine is a Linux application, the command syntax is case-sensitive. The overall syntax is:

```
img [mode]
```

where *mode* is any of the modes described in the sections below.

HINT: Each mode can be abbreviated to the first letter of its name. For example, **`img dump`** can be abbreviated **`img d`**.

- “Help Mode” on page 265
- “Auto Mode” on page 266
- “Make Mode” on page 266
- “Restore Mode” on page 270
- “Dump Mode” on page 275
- “Partition Mode” on page 276
- “Information Mode” on page 278
- “Session (Multicast) Mode” on page 279

Help Mode

Purpose

This is the default mode if you type the **`img`** command with no parameters. Use it to get help with the syntax of the **`img`** command if you don’t have this documentation available.

Format and Function

```
img [h [mode]]
```

where *mode* is the mode whose command syntax you want help with.

Examples

img	Displays a short description of each mode.
img h m	Displays information on how to use the make mode.
img h p	Displays information on how to use the partition mode.

Auto Mode

Purpose

Use this mode to image the computer automatically, based on any applicable NDS policies and settings. The imaging engine runs in this mode if you let the imaging bootup process proceed without interruption, or if you type the command below at the Linux prompt.

Format and Function

```
img a
```

In this mode, the imaging engine queries the imaging server specified in the PROXYADDR environment variable for any work to do. The imaging server checks the relevant NDS policies and settings to determine what imaging tasks should be performed (if any), such as taking or laying down an image. It then instructs the imaging engine to perform those tasks. If any tasks involve storing or retrieving images on other imaging servers, the imaging server refers the imaging engine to those servers to complete those tasks. After the imaging engine has completed its work, it communicates the results to the original imaging server, and the results are logged on that server.

For information on configuring the NDS policies and settings that control what happens in this mode, see Setting Up Imaging Services in Workstation Imaging in *Deployment*.

Make Mode

Purpose

Use this mode to take an image of the computer and store it in a specified location. Normally, all partitions on hard disks and other storage devices (such as Jaz drives) are included in the image, but there are some exceptions noted below.

The image size will correspond roughly to the size of the data in the Windows partitions plus the entire size of any non-Windows partitions (such as NetWare partitions). Linux partitions and Compaq configuration partitions are always excluded. The data from Windows partitions is stored in an intelligent, file-by-file format so you can customize it later using the Image Explorer (IMGEXP.EXE) utility. Non-Windows partitions are stored in a raw, bit-by-bit format that cannot be customized.

Format and Function

The syntax of this mode depends on whether you will store the image locally or on an imaging (proxy) server, as explained in the subsections below:

- "Make Locally" on page 267
- "Make on Proxy" on page 268

Make Locally

Purpose

Use this make mode to take an image of the computer and store it in a partition on a local (writeable) device, such as a hard disk or Jaz drive. For the basic procedure to use this mode, see Setting Up Disconnected Imaging Operations in Setting Up Imaging Services in Workstation Imaging in *Deployment*.

Format and Function

`img ml[`*`partition`*`] `*`filepath`*` [x`*`partition`*`]`

Parameter	Specifies
ml[*partition*]	The partition number (as displayed by **`img dump`**) of the local partition to store the image in. It must be a primary FAT16 or FAT32 partition. This partition is excluded from the image that's created. If you omit the partition number from this parameter, the image is stored in the local ZfD imaging (Linux) partition.
filepath	The image filename, including a .zmg extension and the complete path from the root of the partition. The directories in the path must exist. If the file already exists, it will be overwritten.

Parameter	Specifies
x*partition*	The partition number (as displayed by **`img dump`**) of a local partition to exclude from the image. You can repeat this parameter as needed to exclude multiple partitions. If you omit this parameter, all partitions are included in the image except the one where the image will be stored.

Examples

`img ml8 /imgs/dellnt.zmg`	Takes an image of all partitions except the one in slot 8 and saves the image to imgs/dellnt.zmg in the partition in slot 8. (Assumes slot 8 contains a primary FAT16 or FAT32 partition.)
`img ml /imgs/dellnt.zmg`	Takes an image of all partitions and saves it to imgs/dellnt.zmg in the ZfD imaging (Linux) partition. (Assumes the ZfD partition has been installed.)
`img ml /imgs/dellnt.zmg x2 x3`	Takes an image of all partitions except those in slots 2 and 3 and saves the image to imgs/dellnt.zmg in the ZfD imaging (Linux) partition. (Assumes the ZfD partition has been installed.)

Make on Proxy

Purpose

Use this make mode to take an image of the computer and store it on an imaging (proxy) server. For the basic procedure to use this mode, see Taking an Image of a Workstation Manually in Testing Basic Imaging Operations in Workstation Imaging in *Getting Started*.

Format and Function

`img mp` *`IPaddress filepath`* [*`xpartition`*]

Parameter	Specifies
IPaddress	The IP address or DNS name of the imaging server to store the image on.
filepath	The image filename, including a .zmg extension and the complete path in UNC style. The directories in the path must exist. If the file already exists, the imaging server won't overwrite it unless you enable this behavior in the imaging server's policy in NDS. If no folders are specified in the path, the image is created at the root of the volume or drive where the ZfD imaging server software is installed. **IMPORTANT:** Because Linux doesn't recognize backslashes, you must use forward slashes in the UNC path or enclose the entire path in quotes.
x*partition*	The partition number (as displayed by **`img dump`**) of a local partition to exclude from the image. You can repeat this parameter as needed to exclude multiple partitions. If you omit this parameter, all partitions are included in the image.

Examples

`img mp 137.65.95.127 //xyz_server/sys/imgs/dellnt.zmg`	Takes an image of all partitions and saves it to sys/imgs/dellnt.zmg on xyz_server. (Assumes 137.65.95.127 is the IP address of xyz_server.)
`img mp img.xyz.com //xyz_server/sys/imgs/dellnt.zmg x2 x3`	Takes an image of all partitions except those in slots 2 and 3 and saves the image to sys/imgs/dellnt.zmg on xyz_server. (Assumes img.xyz.com is the DNS name of xyz_server.)

Restore Mode

Purpose

Use this mode to retrieve an image from a specified location and lay it down on the computer.

Normally, if the image to be laid down is a base image (one created previously by the imaging engine), all existing partitions except Linux and Compaq configuration partitions are removed from all local writeable devices (such as hard disks and Jaz drives) before the new image is laid down. When the image is laid down, the sizes of the original partitions from which the image was taken are preserved if possible. If there's insufficient space, the last partition is shrunk to fit unless this would result in data loss, in which case the imaging engine denies the requested operation. If there's extra space left after all partitions in the image have been restored to their original sizes, that space is left unpartitioned.

If the image to be laid down is an add-on image (one produced from an Application object or created by the Image Explorer (IMGEXP.EXE) utility), or if it's a base image and you specify a certain parameter described below, none of the existing physical partitions are removed. Instead, the appropriate partitions are merely updated with the files from the image. The update process does not remove any existing files or overwrite any existing files of the same names if they are newer.

Format and Function

The syntax of this mode depends on whether you will retrieve the image from a local device or from an imaging (proxy) server, as explained in the subsections below:

- "Restore from Local" on page 270
- "Restore from Proxy" on page 273

Restore from Local

Purpose

Use this restore mode to retrieve an image from a local device and lay it down on the computer. For the basic procedure to use this mode, see Setting Up Disconnected Imaging Operations in Setting Up Imaging Services in Workstation Imaging in *Deployment*.

Format and Function

`img rl[partition] filepath [sfileset] [apartition:ppartition]`

Parameter	Specifies
rl[*partition*]	The partition number (as displayed by **`img dump`**) of the local partition to retrieve the image from. It must be a primary FAT16 or FAT32 partition. This partition will not be changed by the imaging operation. If you omit the partition number from this parameter, the image is retrieved from the local ZfD imaging (Linux) partition.
filepath	The filename of the image to retrieve, including the .zmg extension and the complete path from the root of the partition.
s*fileset*	The number of the image fileset (variant) to lay down. Valid values are 1 through 10. For information on creating variants of an image, see Chapter 15, “Preparing Images,” on page 245. If you omit this parameter, fileset 1 is used.

Parameter	Specifies
a*partition*:p*partition*	A mapping between a partition in the image archive (a*partition*) and a target physical partition on the local machine (p*partition*). Use this parameter to selectively restore a specific part of the image to a specific local partition. **IMPORTANT:** If you use this parameter, none of the existing local partitions are removed, and only the target local partition is updated. The update process does not remove any existing files or overwrite any existing files of the same names if they are newer. If you want to remove all existing files from the target partition before updating it, first use **`img pd`** and **`img pc`** to delete and recreate the partition. For a*partition*, use the partition number displayed for the source partition in the Image Explorer (IMGEXP.EXE) utility. For p*partition*, use the partition number displayed by **`img dump`** for the target partition. The target partition must be a Windows partition. You can repeat this parameter as needed to request multiple selective restorations in a single operation. In doing so, you can apply multiple parts of the image to a single local partition, but you can't apply the same part of an image to multiple local partitions in a single operation.

Examples

`img rl8 /imgs/dellnt.zmg`	Removes all existing local partitions except the one in slot 8, retrieves the image from imgs/dellnt.zmg in slot 8, and lays down the partitions and contents of that image on the available local writeable devices. (Assumes there's sufficient local space and that slot 8 contains a primary FAT16 or FAT32 partition.)

img rl /imgs/dellnt.zmg	Removes all existing local partitions, retrieves the image from imgs/dellnt.zmg in the ZfD imaging (Linux) partition, and lays down the partitions and contents of that image on the available local writeable devices (assuming there's sufficient space).
img rl /imgs/dellnt.zmg s2	Removes all existing local partitions, retrieves the image from imgs/dellnt.zmg in the ZfD imaging (Linux) partition, and lays down the partitions and contents of variant 2 of that image on the available local writeable devices (assuming there's sufficient space).
img rl /imgs/dellnt.zmg a2:p1 a3:p1	Retrieves the image from imgs/dellnt.zmg in the ZfD imaging (Linux) partition, updates local partition 1 with the data from partitions 2 and 3 of that image, and leaves the other local partitions unchanged. (Assumes there's sufficient space in local partition 1.)

Restore from Proxy

Purpose

Use this restore mode to retrieve an image from an imaging (proxy) server and lay it down on the computer. For the basic procedure to use this mode, see Putting an Image on a Workstation Manually in Testing Basic Imaging Operations in Workstation Imaging in *Getting Started*.

Format and Function

`img rp` *`IPaddress filepath`* `[s`*`fileset`*`] [a`*`partition`*`:p`*`partition`*`]`

Parameter	Specifies
IPaddress	The IP address or DNS name of the imaging server to retrieve the image from.
filepath	The filename of the image to retrieve, including the .zmg extension and the complete path in UNC style. **IMPORTANT:** Because Linux doesn't recognize backslashes, you must use forward slashes in the UNC path or enclose the entire path in quotes.
s*fileset*	The number of the image fileset (variant) to lay down. Valid values are 1 through 10. For information on creating variants of an image, see Chapter 15, "Preparing Images," on page 245. If you omit this parameter, fileset 1 is used.
a*partition*:p*partition*	A mapping between a partition in the image archive (a*partition*) and a target physical partition on the local machine (p*partition*). Use this parameter to selectively restore a specific part of the image to a specific local partition. **IMPORTANT:** If you use this parameter, none of the existing local partitions are removed, and only the target local partition is updated. The update process does not remove any existing files or overwrite any existing files of the same names if they are newer. If you want to remove all existing files from the target partition before updating it, first use the Partition Mode to delete and recreate the partition. For a*partition*, use the partition number displayed for the source partition in the Image Explorer (IMGEXP.EXE) utility. For p*partition*, use the partition number displayed by **`img dump`** for the target partition. The target partition must be a Windows partition. You can repeat this parameter as needed to request multiple selective restorations in a single operation. In doing so, you can apply multiple parts of the image to a single local partition, but you can't apply the same part of an image to multiple local partitions in a single operation.

Examples

`img rp 137.65.95.127 //xyz_server/sys/imgs/dellnt.zmg`	Removes all existing local partitions, retrieves the image from sys/imgs/dellnt.zmg on xyz_server, and lays down the partitions and contents of that image on the available local writeable devices. (Assumes there's sufficient local space and that 137.65.95.127 is the IP address of xyz_server.)
`img rp img.xyz.com //xyz_server/sys/imgs/dellnt.zmg s2`	Removes all existing local partitions, retrieves the image from sys/imgs/dellnt.zmg on xyz_server, and lays down the partitions and contents of variant 2 of that image on the available local writeable devices. (Assumes there's sufficient local space and that img.xyz.com is the DNS name of xyz_server.)
`img rp img.xyz.com //xyz_server/sys/imgs/dellnt.zmg a2:p1`	Retrieves the image from sys/imgs/dellnt.zmg on xyz_server, updates local partition 1 with the data from partition 2 of that image, and leaves the other local partitions unchanged. (Assumes there's sufficient space in local partition 1 and that img.xyz.com is the DNS name of xyz_server.)

Dump Mode

Purpose

Use this mode to view information about the storage devices and partitions on the computer.

Format and Function

```
img d[g]
```

Parameter	Specifies to
d	List the existing partitions on all local writeable devices, such as hard disks and Jaz drives. For each partition, the type, size, and slot number of the partition are given. **NOTE:** Linux and Compaq configuration partitions are not listed.
g	Display additional information about the geometry (cylinders, heads, and sectors) and capacity of each storage device, including read-only devices such as CD drives.

Examples

`img d`	Lists the current partitions on all local writeable devices.
`img dg`	Lists all storage devices, their geometry and capacity, and the current partitions on the writeable devices.

Partition Mode

Purpose

Use this mode to create, delete, or activate (make bootable) a partition on the computer.

Format and Function

```
img operation
```

where *operation* is one of the following:

Operation	Specifies to
c*partition type* [*size*] [c*clustersize*]	Create a new partition, where: ◆ *partition* is the number of the partition slot (as displayed by **`img d`**) to create the partition in ◆ *type* is FAT12, FAT16, FAT32, NTFS, or Extended ◆ *size* is a valid size for the partition type in MB If you omit this parameter, the largest valid size for the partition type is used, given the available unpartitioned space on the drive. ◆ *clustersize* is the cluster size for an NTFS partition Don't use this parameter unless you have a specific reason to do so. It must be a power of 2 (2, 4, 8, 16, ...128). If you omit this parameter, the imaging engine uses a reasonable cluster size for the NTFS partition size. The new partition is also formatted enough to be recognizable by other operating systems, but you must put a base image in the partition before Windows can store any files in it.
d*partition*	Delete the partition from slot number *partition*. Use **`img d`** to get the slot number.
a*partition*	Activate (make bootable) the partition in slot number *partition*. Use **`img d`** to get the slot number.

Examples

`img pc1 fat16`	Creates a FAT16 partition in slot 1 using all the available unpartitioned space on the drive.

img pc5 fat32 5671	Creates a FAT32 partition in slot 5 using 5,671 MB on the drive.
img pd3	Deletes the partition from slot 3.
img pa5	Activates (makes bootable) the partition in slot 5. (Assumes a partition exists in that slot.)

Information Mode

Purpose

Use this mode to view the following:

- Information about the hardware devices on the computer

  This information is detected during the imaging bootup process. If the imaging engine runs in auto-imaging mode, this information is sent to the imaging server to help determine which image to put on the computer, if necessary.

- The data currently stored in the image-safe area on the computer

  This data is saved off by the imaging agent during each Windows session to ensure that it can be restored after the computer is reimaged. If the computer is brand new and doesn't have Windows yet, an initial set of data is supplied from an NDS policy via the imaging server to the imaging engine when the first base Windows image is laid down.

- Name of the base image that was last laid down on the computer

Format and Function

```
img i [z]
```

Parameter	Specifies to
i	List the detected hardware devices on the computer, including: ◆ CPU chipset ◆ Video adapter ◆ Network adapter ◆ MAC address ◆ Sound card ◆ Hard drive controller ◆ Hard disk capacity ◆ RAM
z	List the data currently stored in the image-safe area on the computer. The items that comprise this data are listed in "Imaging Agent (ZISWIN.EXE)" on page 256. In addition to the image-safe data, the last base image that was laid down on the computer is also listed.

Examples

`img i`	Lists the detected hardware devices on the computer.
`img i z`	Lists the ZfD image-safe data currently stored on the computer and the last base image that was laid down.

Session (Multicast) Mode

Purpose

Use this mode to take an image of one computer and lay it down on multiple other computers simultaneously over the network in a single operation without NDS or ZfD server involvement. For the basic procedure to use this mode, see Chapter 16, "Multicasting Images," on page 251.

For multicasting to work, each participating computer must boot from an imaging device and run the imaging engine in this mode, as explained below. The computer from which the image is taken is called the *master*, and the computers that receive the image are called *slaves*.

NOTE: For multicasting to work properly, the routers and switches on the network must have multicast features configured. Otherwise, multicast packets might not be routed properly.

Format and Function

`img s` *`name`* `[m] [c=`*`count`* `[t=`*`seconds`*`]]`

Parameter	Specifies
name	The name of the multicast session. Each computer joining the session uses the same value for this parameter. **NOTE:** The name must be unique among concurrent multicast sessions. It is hashed by the imaging engine to produce a Class D (temporary) IP address for the multicast session. To facilitate troubleshooting (wire sniffing), all ZfD imaging multicast addresses start with 231. For example, the session name **`doug`** produces the multicast address 231.139.79.72.
m	That this computer is the master and all other participating computers are slaves. If you omit this parameter, the imaging engine waits for a user on one of the computers to press **`m`** to designate that computer as the master.
c=*count*	The number of slaves that must register with the master before imaging will begin. If you omit this parameter, the imaging engine waits for the master user to press **`g`** before starting the imaging process. Once imaging has begun, any slaves attempting to register are denied.
t=*seconds*	The number of seconds the master computer will wait for the next slave to register before starting the imaging process without reaching *count* registered slaves. If you omit this parameter, the imaging process won't start until *count* is reached or the master user presses **`g`**. After that, any slaves attempting to register are denied.

Examples

img s doug	Starts a multicast session named **doug**. Each successive computer that issues this same command before the imaging begins joins the session. Imaging doesn't start until one of the users presses **m** to designate himself as master and presses **g** to start the imaging.
img s doug m	Starts a multicast session named **doug** and designates this computer as the master. Each successive computer that issues **img s doug** before the imaging begins joins the session as a slave. Imaging doesn't start until the master user presses **g**.
img s doug c=5	Starts a multicast session named **doug**. Each successive computer that issues img s doug before the imaging begins joins the session. Imaging doesn't start until one of the users presses **m** to designate himself as master and five other computers register with the master as slaves.
img s doug c=5 t=20	Starts a multicast session named **doug**. Each successive computer that issues img s doug before the imaging begins joins the session. Imaging doesn't start until one of the users presses **m** to designate himself as master and either five other computers register with the master as slaves or more than 20 seconds elapses between any consecutive slave registrations, whichever occurs first.

Imaging Server (IMGSERV.NLM or .DLL or .DLM)

Purpose

The imaging server is a software component of the ZfD server. It enables imaging clients (computers that are booted from an imaging device) to connect with the network to receive imaging services, including:

- Storage or retrieval of an image on a server
- Automatic imaging based on an NDS policy or setting
- Logging of the results of an imaging operation

Location

The imaging server modules are located on a NetWare server in SYS:\SYSTEM or on a Windows server in the folder where NDS is installed (such as C:\NOVELL\NDS).

Format and Function

In most environments, the imaging server starts automatically when you reboot the server after installing ZfD. With NDS eDirectory 8.5 on Windows, you must start the imaging server manually as follows: from the folder where NDS is installed, double-click NDSCONS.EXE > select the IMGSRV.DLM service > click Start. You might also want to click Startup to configure the service to start automatically each time the server reboots.

Once the imaging server has started, you can view information about the status and results of the imaging requests that it has received from imaging clients. A statistical summary of these requests is shown on the server console (NetWare) or in a window accessible from the system tray (Windows). The statistics shown on this screen are explained below. All statistics are reset to zero if you restart the imaging server.

Statistic	Specifies
Update Requests	The number of imaging requests of any kind that have been received by the imaging server since it was started. This includes requests that failed, were denied, or were referred to other imaging servers (see Client Referrals below). Information about each of these requests, such as the source, type, date/time, and results, is logged on the imaging server as explained in "Imaging Server Log (ZIMGLOG.XML)" on page 283.
Images Sent	The number of images that the imaging server has sent to imaging clients since the imaging server was started. This includes only images that were retrieved from this imaging server. See Client Referrals below for more information.
Images Received	The number of new images that have been received and stored on the imaging server since it was started. This includes images that were received through client referrals as explained below.

Statistic	Specifies
Client Referrals	The number of client requests that have been referred (redirected) by the imaging server to other imaging servers since this imaging server was started. Such referrals are made only when the client is running in auto-imaging mode and the imaging server determines from NDS that the image to be created or retrieved is on a different imaging server. **IMPORTANT:** If a client is running in manual imaging mode and it requests to store or retrieve an image on a different imaging server, the request is denied and an error is returned to the client. Referrals are currently supported only when the client is running in auto-imaging mode.

Imaging Server Log (ZIMGLOG.XML)

Purpose

This file is a chronological record of all the imaging requests that have been received by the imaging server since it was installed and first started, including requests merely to log information about imaging operations serviced elsewhere. For each imaging request, the imaging server logs information such as the source, type, date/time, and results of the request.

Location

This file is created on a NetWare server in SYS:\SYSTEM or on a Windows server at the root of the drive where NDS is installed (such as C:\).

Format and Function

This is an XML format file. The oldest imaging request is logged at the top of the file and the most recent at the bottom. The file continues to grow unless you trim it manually (using a text editor) or delete it. It isn't cleared if you restart the imaging server.

Each imaging operation is entered in the log as a group of lines. For example, the log file shown below has two entries: one describing a successful *upload* operation and another describing a failed *download* operation. More information on the meaning of these terms is given below.

```
<ZENImageLog>

<CN=CV7PB00:C0:4F:DC:2A:B5.O=sales>
<Tree>XYZ</Tree>
```

```
<Status>Success</Status>
<Operation>Upload</Operation>
<ImageType>Base Image</ImageType>
<ImagePath>\\XYZ_SERVER\sys\imgs\dell_nt.zmg</ImagePath>
<Timestamp>Wed Jun 21 13:10:05 2000
</Timestamp>
</CN=CV7PB00:C0:4F:DC:2A:B5.O=sales>


<CN=CV7PB00:C0:4F:DC:2A:B5.O=sales>
<Tree>XYZ</Tree>
<Status>Failure</Status>
<ErrorMessage>Unable to find an image to download</ErrorMessage>
<Operation>Download</Operation>
<Timestamp>Wed Jun 21 13:13:17 2000
</Timestamp>
</CN=CV7PB00:C0:4F:DC:2A:B5.O=sales>

</ZENImageLog>
```

The table below describes the various XML elements that comprise the log entries shown above. Each element has an opening and closing tag. For example, <tree> and </tree>. The outermost element contains all the other elements in the entry.

XML Element	Specifies
Outermost_Container	The distinguished NDS name of the workstation that requested the imaging operation. This name is read by the imaging engine from the workstation's image-safe data. If the workstation name isn't found (for example, if the workstation hasn't been registered as an object in NDS), the name of the imaging server that serviced the request is given instead (for example, `XYZ_SERVER`).
Tree	The NDS tree containing the workstation or server specified in the *Outermost_Container* element.
Status	Whether the requested imaging operation succeeded or failed.
ErrorMessage	The reason why the requested imaging operation failed, if applicable.

XML Element	Specifies
Operation	Whether the requested imaging operation was an upload or download attempt. An *upload* is the taking of a client image and storing it on the imaging server or other available (local) medium. A *download* is the retrieval of a client image from the imaging server or local medium and laying it down on the client. **HINT:** Occasionally, you might see an entry in the log that omits the Operation element. Such an entry is typically a follow-up to the previous operation. For example, you might see an entry indicating that a download operation was successful, but the next entry (timestamped a few seconds later and specifying no operation) might indicate that the imaging server failed to get image-safe data from NDS. In this case you could assume that the client that just received the download didn't have its own image-safe data, and so the imaging server was trying to get that data from NDS to apply it to the client.
ImageType	Whether the image that was created or retrieved is a base image or an add-on image. With a base image, all existing partitions and data are removed before the image is laid down. With an add-on image, the existing partitions are left intact and are merely augmented with additional data.
ImagePath	The full path and filename of the image that was created, retrieved, or requested.
Timestamp	The time when the results of the requested imaging operation were logged by the imaging server, including the week day, month, date, 24-hour time (including seconds), and year.

18 Supported Ethernet Cards

The following sections list the Ethernet cards that are supported by ZENworks™ for Desktops (ZfD) for performing network-connected imaging operations on workstations and laptop computers. If your workstation or laptop computer doesn't have one of these cards, you must supply your own Ethernet driver as explained in "DEFAULTDRIVER" on page 263 in Imaging Bootup Parameters (SETTINGS.TXT).

- "Ethernet Cards for a Workstation" on page 287
- "Ethernet Cards for a Laptop Computer (PCMCIA)" on page 288

Ethernet Cards for a Workstation

The following Ethernet cards are supported for standard desktop (non-laptop) workstations:

- 3C501
- Etherlink II, 3c503, 3c503/16
- Etherlink plus 3c505
- Etherlink-16 3c507
- Etherlink III, 3c509 / 3c509B
- 3c515
- 3c590/3c595, 3c592/3c597, 3c900/3c905/3c905B
- AMD Lance (7990, 79C960/961/961 A, Pcnet-ISA), AT1500, HP-J2405A, HP-Vectra On Board Ethernet, NE1500, NE2100
- AT2450, AMD 79C965 (Pcnet-32), AMD 79C970/970A (Pcnet-PCI), AMD 79C971, AMD 79C974
- HP 27245A
- HP EtherTwist, PC Lan+ (27247, 27252A)
- HP 10/100 VG Any Lan Cards (27248B, J2573, J2577, J2585, J970, J973)
- Ether Express

- Ether Express Pro/10
- Ether Express Pro 10/100 B
- NE 1000, NE 2000
- NE2000-PCI
- Racal Interlan ni5010, ni5210, ni6210
- SMC ultra, SMC EtherEZ(8146)
- SMC Ultra32
- SMC 9000/ SMC 91c92/4
- SMC 91c100

Ethernet Cards for a Laptop Computer (PCMCIA)

The following Ethernet cards are supported for laptop (PCMCIA) computers:

Driver	Cards
3c589_cs	◆ 3Com 3c589, 3c589B, 3c589C, 3c589D ◆ 3Com Megahertz 3CCE589E, 3CXE589D, 3CXE589EC ◆ Farallon EtherWave, EtherMac
fmvj18x_cs (x86, ppc)	◆ CONTEC C-NET(PC)C ◆ Eagle NE200 Ethernet ◆ Eiger Labs EPX-10BT, EPX-ET 10BT, EPX-ET 10TZ ◆ Fujitsu FMV-J181, FMV-J182A, FMV-J183 ◆ Fujitsu Towa LA501, FMV-1080, FM50N-183 ◆ Hitachi HT-4840-11 EtherCard ◆ NextCom NC5310 ◆ RATOC REX-9822, REX-5588A/W, REX-R280 ◆ TDK LAC-CD02x, LAK-CD021, LAK-CD022A, LAK-CD021AX, LAK-CD021BX
nmclan_cs	◆ New Media EthernetLAN ◆ New Media LiveWire (*not* LiveWire+)

Driver	Cards
pcnet_cs (A-D)	◆ Accton EN2212, EN2216 EtherCard ◆ Accton SOHO BASIC EN220 ◆ Addtron Ethernet ◆ AIBrain EPCM-T ◆ Allied Telesis CentreCOM CE6001, LA-PCM, LA-PCM V2 ◆ AmbiCom AMB8002, AMB8002T, AMB8010 ◆ AnyCom ECO Ethernet ◆ Apollo RE450CT ◆ Argosy EN210 ◆ Arowana RE 450 Ethernet ◆ Asante FriendlyNet (newer cards seem not to work) ◆ AST 1082 Ethernet ◆ Atelco ethernet ◆ Billionton LNT-10TB, LNT-10TN ◆ California Access LAN Adapter ◆ CeLAN EPCMCIA ◆ CNet CN30BC, CN40BC Ethernet ◆ Compex/ReadyLINK Ethernet Combo ◆ Compex LinkPort Ethernet ◆ COMPU-SHACK BASEline Ethernet ◆ Connectware LANdingGear Adapter ◆ Corega Ether PCC-T, PCM-T ◆ CyQ’ve ELA-010 10baseT ◆ Danpex EN-6200P2 Ethernet ◆ Datatrek NetCard ◆ Dayna Communications CommuniCard E ◆ Digital DEPCM-AA, PCP78-AC Ethernet ◆ Digital EtherWORKS Turbo Ethernet ◆ D-Link DE-650, DE-660 ◆ DynaLink L10C Ethernet

Driver	Cards
pcnet_cs (E-K)	◆ Edimax Technology Ethernet Combo ◆ EFA InfoExpress 205, 207 Combo ◆ Eiger Labs EPX-ET10T2 Combo ◆ ELECOM Laneed LD-CDWA, LD-CDX, LD-CDNIA, LD-CDY, LD-CDF ◆ EP-210 Ethernet ◆ Epson Ethernet ◆ EtherPRIME Ethernet ◆ Explorer NE-10000 Ethernet ◆ EZLink 4109 Ethernet ◆ Fiberline FL-4680 ◆ Gateway 2000 Ethernet ◆ Genius ME3000II Ethernet ◆ Grey Cell Ethernet ◆ GVC NIC-2000P Ethernet Combo ◆ Hamlet LM560 ◆ Hawking PN650TX ◆ Hypertec HyperNet ◆ IBM CreditCard Ethernet Adapter ◆ IC-Card Ethernet ◆ Infotel IN650ct Ethernet ◆ IO DATA PCLA/T, PCLA/TE ◆ Katron PE-520 Ethernet ◆ KingMax Technology EN10-T2 Ethernet ◆ Kingston KNE-PCM/M, KNE-PC2, KNE-PC2T ◆ KTI PE-520 Plus

Driver	Cards
pcnet_cs (L-R)	◆ LANEED LD-CDW Ethernet ◆ LanPro EP4000A ◆ Lantech Ethernet ◆ Level One EPC-0100TB ◆ Linksys EtherCard, EC2T Combo ◆ Logitec LPM-LN10T, LPM-LN10BA, LPM-LN20T Ethernet ◆ Longshine ShineNet LCS-8534TB Ethernet ◆ Macnica ME-1 Ethernet ◆ Maxtech PCN2000 Ethernet ◆ Melco LPC-TJ, LPC-TS, LPC-T, LPC2-T ◆ Microdyne NE4200 Ethernet ◆ Midori LANNER LT-PCMT ◆ NDC Instant-Link ◆ NEC PC-9801N-J12 ◆ Network General "Sniffer" ◆ New Media LanSurfer ◆ Novell/National NE4100 InfoMover ◆ OvisLink Ethernet ◆ Panasonic CF-VEL211P-B ◆ Planet SmartCOM 2000, 3500, ENW-3501-T, ENW-3502-T ◆ Pretec Ethernet ◆ PreMax PE-200 Ethernet ◆ Proteon Ethernet ◆ Psion Gold Card Ethernet ◆ Relia RE2408T Ethernet ◆ Reliasys 2400A Ethernet ◆ RPTI EP400, EP401, 1625B Ethernet

Driver	Cards
pcnet_cs (S-Z)	◆ SCM Ethernet (*not* SMC) ◆ Sky Link Express ◆ SMC 8022 EZCard-10 ◆ Socket Communications EA LAN Adapter ◆ Socket Communications LP-E Ethernet ◆ Socket Communications LP-E CF+ Ethernet ◆ SOHOware ND5120-E Ethernet ◆ SuperSocket RE450T ◆ Surecom Ethernet ◆ SVEC PN605C ◆ Thomas-Conrad Ethernet ◆ TRENDnet Ethernet ◆ Trust Ethernet Combo ◆ UNEX NexNIC MA010 ◆ Volktek NPL-402CT Ethernet
smc91c92_cs	◆ Farallon Enet ◆ Megahertz XJ10BT, XJ10BC, CC10BT Ethernet ◆ New Media BASICS Ethernet ◆ Ositech Four of Diamonds ◆ SMC 8020BT EtherEZ (*not* EliteCard)
xirc2ps_cs	◆ Compaq Ethernet Adapter ◆ Xircom CreditCard CE2, CE IIps, RE-10
3c574_cs Fast Ethernet (10/100baseT) adapters	◆ 3Com 3c574TX, 3CCFE574BT, 3CXFE574BT, 3CCSH572BT, 3CXSH572BT

Driver	Cards
pcnet_cs Fast Ethernet (10/100baseT) adapters	◆ Abocom LinkMate FE1000 ◆ AnyCom ECO Ethernet 10/100 ◆ Apollo Fast Ethernet ◆ COMPU-SHACK FASTline 10/100 ◆ Corega FastEther PCC-TX ◆ D-Link DFE-650 ◆ EXP ThinLan 100 ◆ Fiberline Fast Ethernet ◆ Hamlet FE1000 10/100 ◆ IO DATA PCET/TX ◆ KTI KF-C16 ◆ Laneed LD-10/100CD ◆ LevelOne FPC-0100TX ◆ Linksys PCMPC100 EtherFast, PCM100H1 HomeLink 10/100 ◆ Logitec LPM-LN100TX ◆ Melco LPC2-TX ◆ Microcom TravelCard 10/100 ◆ Micronet EtherFast Adapter ◆ NetGear FA410TXC ◆ New Media LiveWire 10/100 ◆ Planex FNW-3600T ◆ ZONET Fast Ethernet
smc91c92_cs Fast Ethernet (10/100baseT) adapters	◆ Argosy EN220 ◆ Dynalink L100C ◆ Lantech FastNet/TX ◆ Ositech Seven of Diamonds ◆ Melco/SMC LPC-TX ◆ WiseCom WC-PC400

Driver	Cards
xirc2ps_cs Fast Ethernet (10/100baseT) adapters	**NOTE:** Some of these cards seem to misbehave at either 10baseT, 100baseT, or both. The driver doesn't seem to know how to configure the transceiver correctly. ◆ Accton Fast EtherCard-16 ◆ Compaq Netelligent 10/100 ◆ Intel EtherExpress PRO/100 16-bit ◆ Toshiba IPC5008A, Advanced Network 10/100 ◆ Xircom CreditCard CE3-100, CE3B, RE-100
3c575_cb Fast Ethernet (10/100baseT) adapters	◆ 3Com 3c575TX, 3CCFE575BT, 3CXFE575BT, 3CCFE575CT, 3CXFE575CT
epic_cb Fast Ethernet (10/100baseT) adapters	◆ Ositech Seven of Spades CardBus

Driver	Cards
tulip_cb Fast Ethernet (10/100baseT) adapters	◆ Accton EN2220 CardBus ◆ Allied Telesyn AT-2800 ◆ AmbiCom AMB8100 ◆ Apollo FE2000 ◆ Asante FriendlyNET CardBus ◆ Compex Linkport TX ◆ D-Link DFE-660TX ◆ Genius MF3000 (some might not work) ◆ Kingston KNE-CB4TX ◆ Laneed LD-10/100CB ◆ LevelOne FPC-0101TX 10/100Mbps CardBus ◆ Linksys PCMPC200 EtherFast CardBus ◆ OvisLink LFS PCM 32 ◆ SMC EZ CardBus 10/100 Ethernet (some might not work) ◆ SVEC FD606 10/100 Ethernet ◆ TDK NetworkFlyer LAK-CB100X, LAK-CB100AX CardBus ◆ UMAX Technologies UMAX250

V Remote Management

ZENworks™ for Desktops (ZfD) Remote Management gives you the ability to manage remote workstations (called managed workstations) from the management console. Remote Management gives you the ability to do the following:

- Remotely wake up a managed workstation
- Assume control of a managed workstation
- Execute files found on a managed workstation
- Reboot a managed workstation
- Transfer files between the management console and a managed workstation
- Chat with the user at a managed workstation
- Diagnose problems on a managed workstation
- Ping the Remote Management Agent on a managed workstation

Remote Management can save you and your organization time and money. For example, you or your organization's help desk can analyze and remotely fix workstation problems without having to visit the user's workstation; reducing problem resolution times and increasing productivity.

The following sections will help you understand and use ZfD Remote Management:

- Chapter 19, "Understanding Remote Management Components," on page 299
- Chapter 20, "Managing Remote Workstations," on page 311
- Chapter 21, "Diagnostic Information," on page 333

19 Understanding Remote Management Components

You can use ZENworks™ for Desktops (ZfD) to remotely manage Windows* 95, Windows 98, Windows NT*, or Windows 2000 workstations from the management console.

The following sections provide information that will help you understand the functionality of Remote Management components:

- "Understanding Remote Management Agents" on page 299
- "Understanding Remote Management Security" on page 300
- "Understanding Remote Wake Up" on page 300
- "Understanding Remote Control" on page 301
- "Understanding Remote View" on page 301
- "Understanding Remote Execute" on page 301
- "Understanding Remote Diagnostics" on page 301
- "Understanding File Transfer" on page 302
- "Understanding Chat" on page 303
- "Understanding Audit Log" on page 303
- "Monitoring Login and Logout events" on page 308
- "Specifying the Timeout Value for Establishing a Remote Control or Remote View Session" on page 309

Understanding Remote Management Agents

The Remote Management Agent is a ZfD component that is installed on a managed workstation so that the administrator can remotely manage that workstation.

The Remote Management Agent starts automatically when the managed workstation boots up. When you initiate a Remote Management session with a managed workstation, the Remote Management Agent uses NDS® to verify if you have the Remote Management rights. On successful verification the Remote Management session proceeds.

How the Remote Management Agent Uses Protocols

You can use the Remote Management policy to specify the preferred protocol (IP or IPX™) that the agent should use to communicate with the management console during a remote session. If you select a protocol that is not available on that managed workstation, the agent will attempt to use the available protocol. The management console attempts to contact the agent using the network addresses stored within the Workstation object in NDS. It will cycle once through the network addresses trying to communicate with the agent on the managed workstation. For IP addresses in the workstation, the management console attempts to contact the agent using IP. For IPX addresses stored in the Workstation object, the management console attempts to contact the agent using IPX. However, for the management console to communicate with the managed workstation using IPX, ensure that the IP and IPX stacks are installed on the managed workstation. If the IPX stack only is installed, the management console will not be able to communicate with the managed workstation using IPX.

Understanding Remote Management Security

In order for the Remote Management Agent to accept a Remote Management request, the managed workstation must be registered into NDS and be imported as an NDS Workstation object. The Remote Management Agents use NDS authentication to verify that the user requesting to remotely access the managed workstation is authorized to do so. The effective policy settings based on which the administrator performs Remote Management sessions on the managed workstation are taken from the NDS Workstation object and the User object of the user logged in to the managed workstation.

The ZfD management console runs under ConsoleOne™. The Remote Management Agents are NDS authentication-aware and policy-aware and will not allow unauthorized Remote Management sessions to occur.

Understanding Remote Wake Up

Remote Wake Up lets you remotely power up a powered down node in your network (providing that the network card on the node is Wake On LAN enabled). This feature lets the administrator manage nodes during off-hours to minimize the downtime users experience for system maintenance and upgrades. It also facilitates power savings while keeping systems available for

maintenance. For more information, see “Managing a Remote Wake Up Session” on page 314.

Understanding Remote Control

Remote Control lets you control a managed workstation from the management console (ConsoleOne) to provide user assistance and to help resolve workstation problems.

Remote Control establishes a connection between the management console and the managed workstation. With remote control connections, the administrator can go beyond viewing the managed workstation to taking control of it. For more information, see “Managing a Remote Control Session” on page 315.

Understanding Remote View

Remote View lets you view the desktop of the managed workstation from the management console.

Remote View lets you connect with a managed workstation so you can view the managed workstation instead of controlling it. This will help you troubleshoot problems that the user encounters. For example, you can observe how the user at a managed workstation performs certain tasks to ensure that the user performs a task correctly. For more information, see “Managing a Remote View Session” on page 324.

Understanding Remote Execute

Remote Execute lets you run any executable on the managed workstation from the management console. An application can be remotely executed by specifying its executable name in the Remote Execute window (if the program is in the path of the managed workstation) or by entering the complete path of the application (if it is not in the path of the managed workstation). For more information, see “Running an Application on the Managed Workstation” on page 325.

You can determine the value of the path from the Environment window launched from the Diagnostic feature of ZfD. For more information, see “Environment Information” on page 334.

Understanding Remote Diagnostics

Remote Diagnostics helps you shorten problem resolution times and assist users without requiring a technician to physically visit the troubled workstation. This increases user productivity by keeping desktops up and running. Diagnostic information of managed workstations is available over IP only; diagnostics of workstations with pure IPX is not supported. For more information, see “Diagnostic Information” on page 333.

The following table lists the diagnostic information that is available on Windows 95/98 and Windows NT/2000 managed workstations. Diagnostics provide real-time information so the network manager can diagnose workstation problems.

Diagnostic Information	Windows 95/98 managed workstations	Windows NT/2000 managed workstations
Windows Memory	Available	Available
Environment	Available	Available
Event Log	Not Available	Available
WIN32 Processes	Available	Not Available
WIN32 Modules	Available	Not Available
Device Drivers	Not Available	Available
Services	Not Available	Available
NetWare Connections	Available	Available
Novell Client™	Available	Available
Network Protocols	Available	Available
Name Space Providers	Available	Available
Network Drives	Available	Available
Network Open Files	Available	Available
Print Capture	Available	Available

Understanding File Transfer

File Transfer lets you perform file operations between the management console and a managed workstation. To be able to transfer files between the management console and the managed workstation, ensure that the Remote Management Agent is installed on the managed workstation.

Using File Transfer, you can move or copy files between the management console and a managed workstation. You can also rename and delete files, and create directories on the management console and on the managed workstation. From the File Transfer window, you can view the properties of files and directories on the management console and the managed workstation, including size of the file, and the date and time of file creation. File Transfer also lets you open files with the associated application on the management console. For more information, see "Managing a File Transfer Session" on page 327.

IMPORTANT: The File Transfer program does not allow access to non-fixed drives on the managed workstation.

Understanding Chat

Chat is a real-time messaging tool that lets the management console user communicate with a user at the managed workstation. Only a management console user logged in as an administrator can initiate a Chat session. To chat with the user at the managed workstation, you need to ensure that the Remote Management Agent is installed on the managed workstation.

When the management console user initiates a Chat session with the user at the managed workstation, the user at the managed workstation will be prompted for permission to initiate the Chat session. The Chat session begins when the user at the managed workstation provides the permission to initiate the Chat session. During the Chat session, you can copy and paste text in the message area. Either the management console user or the user at the managed workstation can close the Chat session. For more information, see "Managing a Chat Session" on page 330.

Understanding Audit Log

The Windows NT and Windows 2000 event logging mechanism allows applications running on the managed workstation to record events as log files. You can use the Event Viewer to view the event logs. The Event Viewer

maintains Application, Security, and System log files. The events for Remote Management sessions are stored in the Application log file. The managed workstation on which the Remote Management Agent is installed, maintains this log information as an audit log. For more information, see "Viewing the Audit Log of Remote Management Sessions" on page 331.

IMPORTANT: ZENworks 2 stored audit information of Remote Management events in the Security log file. ZfD stores the audit information in the Application log file. You can save the information of previous events using the Save As option from the File menu of the Event Viewer.

The audit log maintains the list of events for each Remote Management session and stores the following details:

- The success or failure of the authentication process
- The permission required status
- The start time or end time of the remote control and remote view session
- The name of the user attempting to remotely manage the workstation
- The domain name and address of the management console accessing the managed workstation
- The remote operation performed on the managed workstation
- The name of the user logged in to the managed workstation
- The event success or failure status, and details for the failure

The following sections contain addtional information:

- "Details of Events in the Audit Log" on page 304
- "Event Log Messages for Remote Management Sessions" on page 306

Details of Events in the Audit Log

The following table explains the information stored by each event during a Remote Management session:

Parameter	Description
Date	Date of the event occurrence.
Time	Timestamp of the event occurrence.
User	Name of the user logged in to the managed workstation.

Parameter	Description
Computer	Name of the computer on which the event occurred.
Event ID	Unique ID assigned to the event.
Source	The source name for the Remote Management audit log is Remote Management Agent.
Type	The type of the event indicates if the particular event was a success, failure, information, warning, or error.
Category	The category lists the different events for the application. The details of an event are in the detailed message for the event. The events for Remote Management Agent are: ◆ Ping Event ◆ Authentication Event ◆ Permission Event ◆ Session Start Event ◆ Session Terminate Event
Operation	The various operations that a management console user can perform on the managed workstation are: ◆ Remote Control ◆ Remote View ◆ Remote Diagnostics ◆ File Transfer ◆ Chat ◆ Remote Execute ◆ Remote Reboot ◆ Ping All events, excluding the Ping event, record the domain name of the administrator who is remotely accessing the managed workstation.
Console Address	IP or IPX address of the workstation that the administrator uses to remotely access the managed workstation.
Console DN	Domain name of the workstation that the administrator uses to remotely access the managed workstation.

Parameter	Description
Local User	Domain name of the user logged in to the managed workstation.
Workstation DN	Domain name of the managed workstation.
Event Message	The message for the event.

Event Log Messages for Remote Management Sessions

Informational and error messages will be recorded for the following events during a Remote Management session:

- “Ping Event” on page 306
- “Authentication Event” on page 306
- “Permission Event” on page 307
- “Session Start Event” on page 307
- “Session Terminate Event” on page 308

You can view the details of events that occurred during a Remote Management session from the Description box in the Event Detail window. For more information about event details, see “Viewing the Audit Log of Remote Management Sessions” on page 331.

Ping Event

The Ping event records if the management console user could verify that the Remote Management Agent is up and running.

Type	Message
Success	Ping was successful
Failure	The managed workstation is being managed by another remote operator

Authentication Event

The Authentication event records if the Remote Management Agent could authenticate the remote user for that operation.

Type	Message
Success	Authentication was successful
Failure	◆ The Remote Management Agent was unable to find the workstation in NDS. Ensure that the workstation is correctly registered. ◆ The remote operator does not have permissions to manage this workstation. ◆ The agent was unable to read NDS for authentication. ◆ The operation is disabled in a policy associated with the workstation of the User object. ◆ There is invalid NDS Authentication information. ◆ An unknown management console tried to establish a Remote Management session.

Permission Event

The Permission event records if the remote user was granted permission for the specified operation.

Type	Message
Information	Permission was requested by the remote operator.
Success	The remote user granted permission for the requested operation.
Failure	◆ The remote user did not grant permission for the requested operation. ◆ The remote user did not grant permission for the requested operation within the specified time interval.

Session Start Event

The Session Start event records the time when a particular session was started.

Type	Message
Information	Session started.

Session Terminate Event

The Session Terminate event details the time at which the session was disconnected, and the reason for terminating the session.

Type	Message
Information	Session terminated normally.
Warning	◆ Remote View session terminated because a mouse or keyboard event was received. ◆ Session terminated because too many negative acknowledgments were outstanding. Check the NetWare® connection. ◆ Session timed out because the management console did not respond.
Error	Remote reboot operation failed, Windows Error Message.

Monitoring Login and Logout events

ZfD takes full advantage of the security functionality of NDS. NDS functionality ensures secure Remote Management sessions when users log out or new users log in to the management console or the managed workstation during a Remote Management session. A remote Management session will terminate, restart, or continue based on the Remote Management security settings for the new user.

Action	Scenario
Session Continue	◆ When the remote management security settings for the new user on the managed workstation are similar to the settings for the current user. ◆ When a new user logs in to the managed workstation and the Audible Signal or Visible Signal settings are different, the session will continue with newer settings.
Session Terminate	◆ When a new user logs in to the management console. ◆ When a new user logs in to the managed workstation and the Remote Control option is disabled.

Action	Scenario
Session Restart	◆ When a new user logs in to the managed workstation and the Screen Blank or Lock Controls settings are different, the session will restart with newer settings. ◆ When a new user logs in to the managed workstation and if permission for a remote session is required from the user at the managed workstation.

Specifying the Timeout Value for Establishing a Remote Control or Remote View Session

If the administrator has the rights to access the managed workstation, the administrator can specify the maximum duration of time to wait for connecting with the managed workstation to start a Remote Control or a Remote View session. The default timeout value is 5 minutes. The administrator can change the default timeout value and choose a value between the range of 1 minute and 30 minutes.

When the administrator initiates a Remote Control or Remote View session, the management console attempts to establish connection with the managed workstation. If the Remote Management Agent is up and running on the managed workstation and if the network link between the management console and the managed workstation is up, the connection for Remote Control or Remote View will be established and the session will proceed.

To specify the timeout value for establishing a Remote Control or Remote View session:

1. From the management console, right-click the User object for the management console user.
2. Click Properties > Remote Management.
3. Click the General tab.
4. In the Remote Control/View Timeout for Console User text field, specify the maximum duration for the management console to try and establish a Remote Control or Remote View session with the managed workstation.

   IMPORTANT: The timeout value set in the User object or associated policy for the User object is applicable to the management console user, not to the managed workstation user. All other security settings in the Remote Management property page are applicable only to the managed workstation user.

20 Managing Remote Workstations

The following sections provide information that will help you effectively manage Remote Management sessions:

- “Using the Remote Management Agents” on page 311
- “Managing a Remote Wake Up Session” on page 314
- “Managing a Remote Control Session” on page 315
- “Managing a Remote View Session” on page 324
- “Running an Application on the Managed Workstation” on page 325
- “Viewing Diagnostic Information for a Managed Workstation” on page 326
- “Managing a File Transfer Session” on page 327
- “Managing a Chat Session” on page 330
- “Viewing the Audit Log of Remote Management Sessions” on page 331

Using the Remote Management Agents

You will be able to access managed workstations if you have installed the Remote Management Agent on the managed workstation. For more information, see “Understanding Remote Management Agents” on page 299.

The following sections explain how you can use the Remote Management Agent during remote sessions:

- “Pinging the Remote Management Agent from the Management Console” on page 311
- “Shutting Down the Remote Management Agent” on page 312
- “Reloading the Remote Management Agent” on page 313
- “Uninstalling the Remote Management Agent using the Novell Application Launcher” on page 313

Pinging the Remote Management Agent from the Management Console

You can ping the Remote Management Agent on the managed workstation before you begin a Remote Management session. The Ping window displays the response received from the Remote Management Agent based on the packets sent from the management console, which helps you determine if the Remote Management Agent is loaded on the managed workstation.

To ping the managed workstation from the management console:

1. Right-click the managed workstation from the management console.
2. Click Actions > Ping Remote Management Agent.

The following table explains the results displayed in the Ping window:

Statistic	Explanation
Total Packets Sent	Total number of packets sent by the management console to the Remote Management Agent on the managed workstation
Packets Received	Total number of packets received by the management console from the Remote Management Agent
Packets Per Second	Number of packets sent by the management console to the Remote Management Agent per second
Response Ratio	Number of packets received by the management console from the Remote Management Agent in response to the packets sent by the management console

Shutting Down the Remote Management Agent

You can shut down the Remote Management Agent during a remote session. When you shut down the Remote Management Agent, the remote session stops. To start another remote session, you will need to reload the Remote Management Agent. For more information, see “Reloading the Remote Management Agent” on page 313.

To shut down the Remote Management Agent from a Windows 95/98 managed workstation:

1. Right-click the Remote Management icon.

2 Click Shut Down Agent.

To shut down the Remote Management Agent from a Windows NT/2000 managed workstation:

1 From the Control Panel, click Services.

2 Select Remote Management.

3 Click Stop.

IMPORTANT: You will be able to stop the Remote Management Agent on Windows NT/2000 only if you have the rights to stop the Windows NT/2000 service.

Reloading the Remote Management Agent

During ZfD installation, the Remote Management Agent is installed on the managed workstation and started automatically when the managed workstation starts up. If you shut down the Remote Management Agent during a remote session, the remote session stops. To start another remote session, you need to reload the Remote Management Agent on the managed workstation.

To reload the Remote Management Agent on Windows 95/98:

1 Run ZENRC32.EXE from the \NOVELL\ZENRC\ directory.

To reload the Remote Management Agent on Windows NT/2000:

1 From the Control Panel, click Services.

2 Select Remote Management.

3 Click Start.

IMPORTANT: You will be able to load the Remote Management Agent on Windows NT/2000 only if you have the rights to start the Windows NT/2000 service.

Uninstalling the Remote Management Agent using the Novell Application Launcher

You can uninstall the agent by adding the Remote Management Uninstall Application object to the Novell Application Launcher™ and associating the Application object with the managed workstation.

IMPORTANT: For Windows NT/2000 managed workstations, you must associate the Application object with the Workstation object or Container of the Workstation

object. You will not be able to launch the Application object if you associate it with a User object.

To uninstall the Remote Management Agent using the Application Launcher:

1. From the management console, right-click a managed workstation.
2. Click Details > Applications.
3. Click Add > Remote Management Uninstall.
4. Select an association for Remote Management Uninstall, from the options explained in Installing the Remote Management Agent using the Application Launcher in Remote Management in *Getting Started*.

Managing a Remote Wake Up Session

The Remote Wake Up feature supports Magic Packet technology. When a powered off, Wake on LAN-enabled node receives the magic packet, the system will boot up.

You can use ZfD to remotely wake up Windows 95, Windows 98, Windows NT 4.0, and Windows 2000 nodes.

For an administrator to be able to wake up a remote node, the following requirements must be met:

- The remote node has a network card that supports Wake-on-LAN. Additionally, ensure that you have enabled the Wake-on-LAN option in the BIOS setup of the managed workstation.
- The remote node is in a soft-off power state. In the soft-off state, the CPU is powered off and a minimal amount of power is utilized by its network interface card. Unlike the hard-off state, in the soft-off state the power connection to the machine remains switched on when the machine is shut down.

  NOTE: Some operating systems like Windows NT and Windows 2000 do not put the machine in the soft-off state even after the corresponding shut down command is executed. When a user selects the Shut Down option from the System Menu of Windows NT/2000, the machine is not shut down automatically, the user is prompted to either reboot or shut down the machine. For Remote Wake Up to work, the user should manually shut down the machine upon this prompt.
- The remote node has been scanned at least once by the *Scanner* that ships with ZfD.
- The Wake on LAN Status Agent is properly installed on the remote node.

HINT: The Wake on LAN Status Agent is installed on the managed workstation during Remote Management Agent installation.

- The routers connecting the management console and the remote node are configured to forward subnet-oriented broadcasts.
- The network link between the management console and the remote node is established.

To wake up a remote node:

1. Right-click the Workstation Object from the management console.
2. Click Actions > Remote Wake Up.

Managing a Remote Control Session

ZfD lets you remotely control a managed workstation. For more infromation, see "Understanding Remote Control" on page 301. You can effectively manage a Remote Control session by performing the following tasks with the Viewing window control options, the Viewing window toolbar buttons, and the Remote Management icon options:

- "Starting a Remote Control Session" on page 315
- "Enabling the Wallpaper on the Managed Workstation" on page 315
- "Controlling the Display of the Viewing Window" on page 316
- "Using the Viewing Window Accelerator Keys" on page 317
- "Using the Toolbar Buttons on the Viewing Window" on page 319
- "Using the Remote Management Icon" on page 320
- "Obtaining Information About Remote Management Sessions" on page 321
- "Stopping a Remote Control Session from the Managed Workstation" on page 324

Starting a Remote Control Session

Before you start a Remote Control session with the managed workstation, make sure that the appropriate Remote Management Agent is loaded on the managed workstation.

To start a Remote Control session:

1. Right-click the managed workstation from the management console.
2. Click Actions > Remote Control.

You can configure the timeout value for establishing a Remote Control session. For more information, see “Specifying the Timeout Value for Establishing a Remote Control or Remote View Session” on page 309.

Enabling the Wallpaper on the Managed Workstation

When an administrator initiates a Remote Control or Remote View session with the managed workstation, any wallpaper displayed on the desktop of the managed workstation will be suppressed. This feature reduces the response time from the managed workstation for requests from the management console because less traffic is generated over the network while the wallpaper is suppressed.

You can configure the control parameter for this option to change the default settings and enable the display of the wallpaper on the managed workstation. When you terminate the Remote Control or Remote View session, the suppressed wallpaper will be restored.

NOTE: On Windows 95/98 managed workstations, the Suppress Wallpaper option is enabled only if you log in to the workstation as a Windows User.

To enable the display of suppressed wallpaper on the managed workstation:

1. Click the Remote Management icon at the top-left corner of the Viewing window > click Configure.
2. Deselect the Suppress Wallpaper control option.

Controlling the Display of the Viewing Window

You can control the display of the managed workstation by using the Viewing window control options.

To enable control options:

1. Click the Remote Management icon at the top-left corner of the Viewing window.
2. Click Configure.
3. Select the control options you want to enable for the remote session.

The following table explains the options you can use to control the display of the Viewing window.

Option	Description
Warn me on Screen Blanking	Informs the user at the management console before the managed workstation screen is blanked.
Accelerator Keys Enabled	Enables the accelerator keys on the management console so that default accelerator key sequences are always available during the remote session. For details, see "Using the Viewing Window Accelerator Keys" on page 317.
System Key Pass Through	Passes Alt-key sequences on the management console to the managed workstation. During a Remote View session, the System Key Pass-Through option is not enabled.
Suppress Wallpaper	Suppresses any wallpaper displayed on the managed workstation.
Force 16 Color Viewing	Forces the use of 16-color palette on the managed workstation during a Remote Management session. This enhances the Remote Management performance. Use this option only if you are performing a Remote Management session over a slow WAN.

Using the Viewing Window Accelerator Keys

You can use accelerator keys to control the display of the Viewing window. Default accelerator key sequences are assigned to each accelerator key option. The Accelerator Keys dialog box displays the default key sequence in the edit field of each accelerator key option. You can define a custom accelerator key sequence to change the default sequence. For more informatin, see "Defining a Custom Accelerator Key Sequence" on page 318.

To open the Accelerator Keys dialog box:

1. Click the Remote Management icon at the top-left corner of the Viewing window.
2. Click Accelerator Keys.

The following table explains the Accelerator Key options you can use to control the display of the Viewing window:

Option	Default Keystroke	Description
Full Screen Toggle	Ctrl+Alt+M	Applicable only if the color resolution settings on the management console and managed workstation are similar. Sizes the Viewing window to the size of your screen without window borders.
Refresh Screen	Ctrl+Alt+R	Refreshes the Viewing window.
Restart Viewer	Ctrl+Alt+T	Re-establishes connection with the managed workstation.
System Key Pass	Ctrl+Alt+S	Passes Alt-key sequences on the management console to the managed workstation.
Accelerator Key Enable	Ctrl+Alt+H	Enables the use of accelerator key sequences.
Stop Viewing	Left-Shift+Esc	Closes the Viewing window.

To enable the Accelerator Keys option:

1. Click the Remote Management icon at the top-left corner of the Viewing window.
2. Click Configure.
3. Select Accelerator Keys Enable.

Defining a Custom Accelerator Key Sequence

Default keystrokes assigned to the accelerator key options are displayed in the edit field to the right of each accelerator key option. You can change the accelerator key sequence and define a custom accelerator key sequence if you do not want to use the default keystroke.

To define a custom accelerator key sequence:

1. Click the Remote Management icon at the top-left corner of the Viewing window.
2. Click Accelerator Keys.

3. Click the edit field of the accelerator key option where you want to define a custom accelerator key sequence.
4. Press the new accelerator key sequence.

IMPORTANT: The shift keys are left-right sensitive, and are indicated in the Control Options dialog box as Lshift and Rshift.

Using the Toolbar Buttons on the Viewing Window

The following table describes the toolbar options in the Viewing window:

Button	Description	Key Function
Screen Blank	Blanks the screen of the managed workstation.	Displays only if the Allow Blanking User's Screen option is enabled in the security settings. Blanks the screen at the managed workstation. When the administrator selects this option, the screen of the managed workstation will be blacked-out and the operations performed by the administrator on the managed workstation will not be visible to the user at the managed workstation. Not supported over certain display adapters. Refer to the ZfD Readme for a list of display adpaters that do not support this feature.
Lock Controls	Locks the keyboard and mouse controls of the managed workstation.	Locks the keyboard and mouse controls at the managed workstation. When the administrator selects this option, the user at the managed workstation will not be able to use the keyboard and mouse controls of the managed workstation.
Start	Sends the Ctrl+Esc keystroke to the managed workstation.	Opens the taskbar with the Start button on Windows 95/98/NT/2000 workstations.

Button	Description	Key Function
Application Switcher	Sends the Alt-tab key sequences to the managed workstation.	Switches application on managed workstations. If this option is on, you will need to press only the Tab key on the management console to select applications, and then switch the button off.
CTRL-ALT-DEL/ Reboot	Sends the Ctrl+Alt+Del keystroke to the managed workstation.	This button is labelled Reboot when the managed workstation is Windows 95/98. Restarts the Windows 95/98 workstations. This button is labelled CTRL-ALT-DEL, when the managed workstation is Windows NT/2000. Displays the Security window on Windows NT/2000.
System Key Pass-Through	Sets the system key pass to On or Off.	Passes Alt-key sequences on the management console to the managed workstation. Certain key sequences such as Ctrl+Esc, Alt+Tab, Ctrl+Alt+Del, and Alt+PrintScreen are not allowed even when the System Key Pass-Through is set to On. However, you can use the toolbar buttons on the Viewing window for the Ctrl+Esc, Alt+Tab, and Ctrl+Alt+Del keystrokes.
Navigate	Represents the desktop of the managed workstation on the management console.	Displays the different areas of the desktop of the managed workstation. You can move the red frame on the Viewing window to focus on a particular area of the managed workstation desktop.

IMPORTANT: During a Remote View session, only the Navigate button is displayed in the Viewing window.

Using the Remote Management Icon

You can manage a remote session from the managed workstation using the Remote Management icon options. The Remote Management icon will be displayed in the taskbar of the Windows 95, Windows 98, Windows NT 4.0, and Windows 2000 workstations.

This icon indicates that the Remote Management Agent is loaded on the managed workstation. If the Remote Management Agent is loaded and the Remote Management icon is not displayed in the taskbar, Task List, or the desktop, it indicates that you have disabled the display option in the Remote Management Policy settings. After the Remote Management Agent is loaded, an administrator with sufficient rights can start a remote session on the managed workstation.

The user at the managed workstation can right-click the Remote Management icon and choose from the following options:

Option	Description
Terminate Session	Disconnects and closes the remote session on the managed workstation and displays a message on the management console indicating that the remote session is closed.
Shut Down Agent	Shuts down the Remote Management Agent on the managed workstation. To start another remote session, the Remote Management Agent should be reloaded on the managed workstation. For details, see “Reloading the Remote Management Agent” on page 313. This option is not enabled for Windows NT/2000 managed workstations. The Remote Management Agent runs as a service on Windows NT/2000 and can be shut down from the Windows NT/2000 Services dialog box.
Information	Displays information such as who is accessing the managed workstation for the remote session, security settings, and the protocol in use for the remote session. For details, see “Obtaining Information About Remote Management Sessions” on page 321. You can right-click or double-click the Remote Management icon to view the Information window.
Help	Displays the Remote Management Agent help file.

Obtaining Information About Remote Management Sessions

Using the Information window, the user at the managed workstation can view details, such as who is accessing the managed workstation for a remote session, the security settings, and the protocol in use for the remote session.

To view information about remote sessions:

1. Right-click the Remote Management icon.
2. Click Information.

Once you have opened the Information window, you can view different kinds of information about remote sessions on the managed workstation. See the following sections for details:

- "Obtaining General Information" on page 322
- "Obtaining History Information" on page 322
- "Obtaining Security Information" on page 323

Obtaining General Information

The following table explains the general information you can obtain about Remote Management sessions from the Information window:

Parameter	Description
Status	Displays the name of the administrator who is performing a remote session on the managed workstation, with the date and time the managed workstation was accessed.
Protocol	Displays the protocol that the Remote Management Agent uses to communicate with the management console during a remote session.
Optimization	Displays if optimization is enabled or disabled for the Remote Management session. The Remote Management Agent performance will be optimized if the video card on the managed workstation is compatible with the performance enhancement driver that is installed during Remote Management Agent installation.

Obtaining History Information

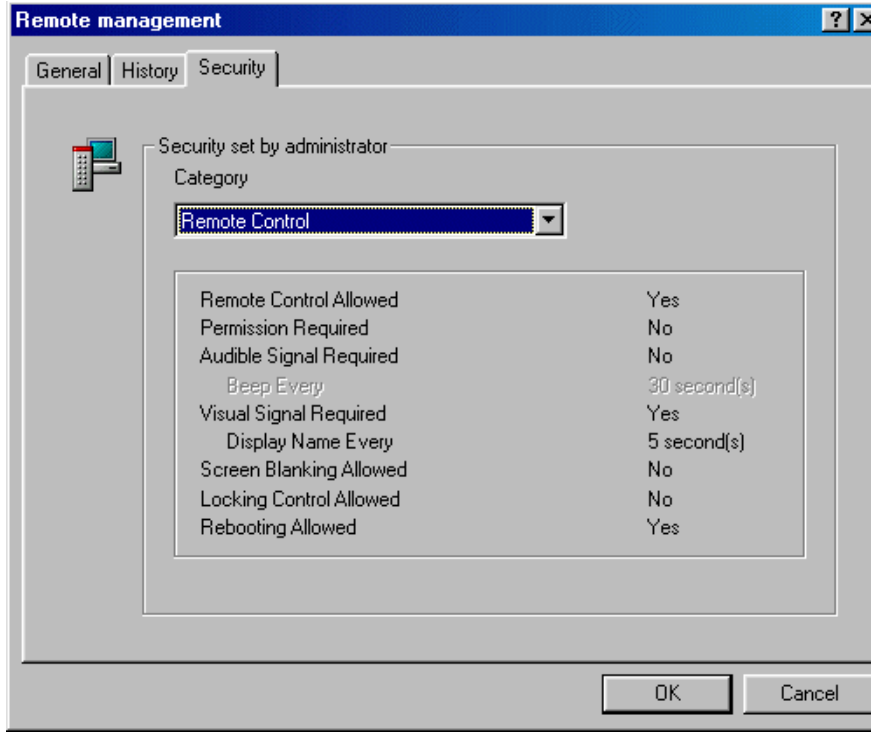
The following table explains the history information you can obtain about Remote Control and Remote View sessions from the Information Window:

Parameter	Description
Operation	Displays a list of the last ten remote sessions.
Initiator	Displays the name of administrator who initiated the remote session.
Start Date/Time	Displays the date and time when the remote session was initiated.

Obtaining Security Information

The Security Information Window displays information based on the following categories of remote sessions:

- Remote Control
- Remote View
- File Transfer
- Remote Execute
- Others (Chat and Diagnostics)



The administrator can use the Remote Management Policy pages to change the settings for each parameter.

Stopping a Remote Control Session from the Managed Workstation

To stop a Remote Control session from the managed workstation:

1. Right-click the Remote Management icon.
2. Click Terminate Session.

Managing a Remote View Session

You can use ZfD to remotely view the managed workstation. For more information, see “Understanding Remote View” on page 301. The following sections explain the tasks you can perform to effectively manage a Remote View session:

- “Starting a Remote View Session” on page 324

- "Controlling the Display of the Viewing Window During a Remote View Session" on page 324
- "Stopping a Remote Control Session from the Managed Workstation" on page 324

Starting a Remote View Session

To start a Remote View session:

1. Right-click the managed workstation from the management console.
2. Click Actions > Remote View.

You can configure the timeout value for establishing a remote view session. For more information, see "Specifying the Timeout Value for Establishing a Remote Control or Remote View Session" on page 309.

Controlling the Display of the Viewing Window During a Remote View Session

When you initiate a Remote View session, a representation of the desktop of the managed workstation will be displayed on your screen in a separate window called the Viewing window. The following table lists the control options and toolbar button that you can use during a Remote View session. For a description of the options and button, see "Controlling the Display of the Viewing Window" on page 316 and "Using the Toolbar Buttons on the Viewing Window" on page 319.

Options	Option Name
Control Options	Warn me on Screen Blanking
	Accelerator Keys Enabled
	Force 16 Color Viewing
Toolbar Button	Navigate

Stopping a Remote View Session from the Managed Workstation

To stop a Remote View session from the managed workstation:

1. Right-click the Remote Management icon.

2 Click Terminate Session.

Running an Application on the Managed Workstation

You can remotely run executables on the managed workstation using the Remote Execute feature of ZfD. For more information, see “Understanding Remote Execute” on page 301.

To execute an application program on a managed workstation:

1 Right-click the managed workstation from the management console.

2 Click Actions > Remote Execute.

3 Enter the command line in the Remote Execute window.

Specify the complete path of the application if the application is not in the path of the managed workstation.

If you do not specify the extension of the file you want to execute at the managed workstation, Remote Execute will append the .EXE extension.

4 Click Execute.

Enter the name of the application or the parameter within double-quotes if the application or parameter has a space character. Following are a few examples:

```
"My Wordpad"

"C:\Program Files\Accessories\My Wordpad"

"C:\Program Files\Accessories\MyWordpad" "C:\myfile.txt"

"C:\Program Files\Accessories\My Wordpad" C:\myfile.txt

Wordpad
```

Viewing Diagnostic Information for a Managed Workstation

You can view diagnostic information that will help you analyze problems at the managed workstation. For more information, see “Understanding Remote Diagnostics” on page 301.

To view Diagnostic information:

1 Right-click the managed workstation from the management console.

2 Click Actions > Diagnostics.

2a To view the Windows Memory window, expand Diagnostics folder > Operating System folder > Memory folder > click Windows Memory.

For more information, see “Windows Memory Information” on page 334.

2b To view the Environment window, expand Diagnostics folder > Operating System folder > click Environment.

For more information, see “Environment Information” on page 334.

2c To view the Event Log window, expand Diagnostics folder > Operating System folder > click Event Log > Security, System, or Application.

Click an event row in the Event Log table to view a description of the event.

For more information, see “Event Log Information” on page 335.

2d To view the Device Drivers window, expand Diagnostics folder > Operating System folder > click Device Drivers.

For more informatin, see “Device Drivers Information” on page 336.

2e To view the Services window, expand Diagnostics folder > Operating System folder > click Services.

For more information, see “Services Information” on page 336.

2f To view the WIN32 Processes window, expand Diagnostics folder > Operating System folder > click WIN32 Processes.

For more information, see “WIN32 Processes Information” on page 336.

2g To view the WIN32 Modules window, expand Diagnostics folder > Operating System folder > click WIN32 Processes.

Double-click a row entry to view the associated WIN32 Modules.

For more information, see “WIN32 Modules Information” on page 337.

2h To view the NetWare Connections window, expand Diagnostics folder > Network folder > click NetWare Connections.

For more information, see “NetWare Connections Information” on page 337.

2i To view the Network Protocols window, expand Diagnostics folder > Network folder > click Network Protocols.

For more information, see “Network Protocols Information” on page 339.

2j To view the Name Space Providers window, expand Diagnostics folder > Network folder > click Name Space Providers.

For more information, see “Name Space Providers Information” on page 340.

2k To view the Network Drives window, expand Diagnostics folder > Network folder > click Network Drives.

For more information, see “Network Drives Information” on page 340.

2l To view the Network Open Files window, expand Diagnostics folder > Network folder > click Open Files.

For more information, see “Network Open Files Information” on page 342.

2m To view the Novell Client window, expand Diagnostics folder > Network folder > click Novell Client.

For more information, see “Novell Client Information” on page 338.

2n To view the Print Capture window, expand Diagnostics folder > Network folder > click Print Capture.

For more information, see “Print Capture Information” on page 343.

HINT: You can use the Edit menu options to copy all or selected diagnostic information from the diagnostics windows to a text-editor for later analysis.

Managing a File Transfer Session

ZfD lets you transfer files between the management console and a managed workstation. For more information, see “Understanding File Transfer” on page 302. Before you begin a File Transfer session, ensure that you have uninstalled any third-party File Transfer service from the managed workstation.

The following sections explain how you can use File Transfer and the options that are available for working with files from the File Transfer window:

- “Starting a File Transfer Session” on page 328
- “Using File Transfer Window Controls” on page 328

Starting a File Transfer Session

To start a File Transfer session:

1. Right-click the managed workstation from the management console.
2. Click Actions > File Transfer.

Using File Transfer Window Controls

The left pane of the File Transfer window shows the files in the current folder on the management console and the right pane shows the files on the managed workstation. The following table explains the function of the File Transfer controls:

Menu Option	Toolbar Option	Description
File > Open		Opens the selected file in Notepad or Wordpad at the management console. Opens the folder with the list of files at the management console.
File > Open with		Opens the file at the management console with the specified application.
File > New Folder		Creates the folder with the specified name.
File > Delete		Deletes the selected files. Deletes the folder if the folder selected from the management console is empty.
File > Rename		Renames the selected file.
File > Properties		Displays the properties of a selected file or folder, such as size of the file and the date and time of last modification.

Menu Option	Toolbar Option	Description
File > Upload		Moves files from the management console to the managed workstation.
File > Download		Moves files from the managed workstation to the management console.
File > Exit		Closes the File Transfer window.
Edit > Cut		Transfers the selected files to the Clipboard.
Edit > Copy		Copies the selected files to the Clipboard.
Edit > Paste		Pastes the selected files from the Clipboard to the current location.
Edit > Select All		Selects all the files in the current pane.
Edit > Cancel All		Deselects all the files in the current pane.
View Refresh		Updates the display in the Operator Station pane and Target Station pane
Help	Help	Displays help for this window.
Up One Level Folder button		Moves one level up in the directory tree. Right-click the file or folder to view the list of available menu options.
Operator Station Pane		The left pane of the File Transfer window shows the files in the current folder on the management console.
Target Station Pane		The right pane of the File Transfer window shows the files in the current folder on the managed workstation.

Managing a Chat Session

The Chat feature lets you communicate with the user at the managed workstation. For more information, see “Understanding Chat” on page 303. The following sections explain how you can manage Chat sessions using the Chat window:

- “Chatting with a User at the Managed Workstation” on page 330
- “Editing the Messages in the Chat Window on the Management Console” on page 330

HINT: Ensure that the Remote Management Agent that ships with ZfD is installed on the managed workstation. If you have installed an earlier version of the Remote Management Agent, you will be prompted to upgrade the agent before the Chat program loads.

Chatting with a User at the Managed Workstation

To chat with the user at the managed workstation:

1. Right-click the recipient’s managed workstation from the management console.
2. Click Actions > Chat.

   To proceed with the Chat session, the user at the managed workstation must accept the Chat session.
3. Type a message.
4. Click File > Exit to exit the Chat window.

Editing the Messages in the Chat Window on the Management Console

To clear the contents of the text boxes during a Chat session:

- To clear the contents of the Input text box, click Edit > Clear Input.
- To clear the contents of the Response text box, click Edit > Clear Response.
- To clear the contents of the both text boxes, click Edit > Clear All.

To insert existing text into Chat window text boxes:

- To copy the selected text to the Windows Clipboard, click Edit > Copy.
- To add copied text to a text box, Click Edit > Paste.

Viewing the Audit Log of Remote Management Sessions

ZfD records log information on a Windows NT/2000 managed workstation. For more information, see “Understanding Audit Log” on page 303.

To view the audit log of Remote Management sessions:

1. Click Start > Programs > Administrative Tools > Event Viewer.
2. Click Log > Application.
3. Double-click the event associated with the source Remote Management Agent.

   HINT: To view only the events pertinent to the Remote Management Agent, choose Remote Management Agent from the source drop-down list in the Filter dialog box.

ZfD provides remote diagnostics of workstations. Remote diagnostics displays the event log information of Windows NT/2000 managed workstations. You can also view the audit log for Remote Management using the Event Log window. For more information, see “Event Log Information” on page 335.

21 Diagnostic Information

You can diagnose the managed workstation and obtain information that will help you analyze problems at the managed workstation. For more information, see “Understanding Remote Diagnostics” on page 301.

You can view real-time managed workstation diagnostic information from the management console. For more information, see “Viewing Diagnostic Information for a Managed Workstation” on page 326.

Before you begin to obtain diagnostic information, ensure that the Remote Management Agent is installed on the managed workstation. During Remote Management Agent installation, the Diagnostic Agent is also installed on the managed workstation, which runs automatically when the managed workstation boots up. When the management console user requests diagnostic information from the managed workstation, the Diagnostic Agent on the managed workstation procures the requested information and provides it to the Remote Management Agent, which then makes it available to the management console.

The following sections describe the diagnostic information you can obtain using ZdD:

- “Windows Memory Information” on page 334
- “Environment Information” on page 334
- “Event Log Information” on page 335
- “Device Drivers Information” on page 336
- “Services Information” on page 336
- “WIN32 Processes Information” on page 336
- “WIN32 Modules Information” on page 337
- “NetWare Connections Information” on page 337
- “Novell Client Information” on page 338
- “Network Protocols Information” on page 339
- “Name Space Providers Information” on page 340
- “Network Drives Information” on page 340
- “Network Open Files Information” on page 342

◆ “Print Capture Information” on page 343

Windows Memory Information

On Windows 95/98 and Windows NT/2000 managed workstations, the Windows Memory window displays the percentage of memory in use, physical memory, paging details, and free space details.

You can view the following information about Windows memory on a managed workstation:

Field	Description
Memory Load (%)	Percentage of memory utilization. Zero percentage memory indicates memory usage is nil; 100% indicates that all the available memory is in use.
Total Physical Memory (MB)	Total physical memory in MB.
Free Physical Memory (MB)	Amount of available physical memory in MB.
Total Paging File Size (MB)	Total number of MB that can be stored in the paging file. This number does not indicate the actual physical size of the paging file on the managed workstation.
Free Space in Paging File (MB)	Number of MB available in the paging file.
Total Address Space (MB)	Total number of MB described in the user mode portion of the virtual address space of the calling process.
Free User Bytes (MB)	Number of MB in unreserved and uncommitted memory of the user address space of the calling process.

Environment Information

The Environment window displays the variables set at the managed workstation. You can view the Environment information on Windows 95/98 and Windows NT/2000 managed workstations.

The following table explains each field in the Environment window:

Field	Description
Variables	Environment variable name.
Value	Value of the variable or the path.

Event Log Information

Event logging in Windows NT/2000 provides a standard, centralized way for applications and the operating system to record important software and hardware events. Event logging provides a means to merge events from various sources into a single informative story. The event log diagnostics help the administrator view the System, Security, and Application event logs. You can view the Event Log Information on Windows NT/2000 managed workstations.

The following table explains each field in the Event Log window:

Field	Description
Event Generated Date	Date on which the entry was submitted (MM/DD/YYYY).
Event Generated Time	Time at which the entry was submitted (HH:MM:SS).
Event ID	Identifies the event specific to the source that generated the event log entry.
Event Generated Type	Classification of the type as Error, Warning, Information, Success, or Failure.
Event Generated Category	Subcategory for the event. This subcategory is source specific.
Source Name	Name of the source (application, service, driver, subsystem) that generated the entry.
Description	Details of the event.
Computer Name	Name of the computer that generated the event.

Device Drivers Information

The Device Drivers window displays information about the device drivers installed on Windows NT/2000 managed workstations. You can use the information in this window to determine whether the workstation has the required drivers loaded and their status.

The following table explains each field in the Device Driver window for Windows NT/2000 managed workstations:

Field	Description
Name	Name of the device driver.
State	Indicates if the device driver is Stopped or Running.

Services Information

The Services window indicates which services are available on Windows NT/2000 managed workstations, and lists the state of each service.

The following table describes the fields in the Services window:

Item	Description
Service Name	List of services available on the workstation.
State	Indicates if the service is Stopped or Running.

WIN32 Processes Information

Diagnostic information about processes is available on Windows 95/98 managed workstations.

To view the WIN32 Modules associated for a particular Windows 32-bit process, double-click the row entry in the WIN32 Processes window.

The following table describes the fields in the WIN32 Processes window:

Field	Description
Path	Path and filename of the executable file for the process.

Field	Description
PID	Processor identifier.
PPID	Parent process identifier.
No. of Threads	Number of execution threads started by the process.
Usage Count	Number of references to the process. A process exists as long as its usage count is non-zero. When the usage count becomes zero, the process terminates.

WIN32 Modules Information

The WIN32 Modules window displays the list of modules associated with a specified process on Windows 95/98 managed workstations.

The following table describes the fields in the WIN32 Modules window:

Field	Description
Module ID	Module identifier in the context of the owning process.
Global Usage Count	Global usage count on the module.
Process Usage Count	Module usage count in the context of the owning process.
Module Path	Location of the module.
Module Size (KB)	Size of the module in KB.

NetWare Connections Information

The NetWare Connections window displays information about all current connections for the Novell Client. It also indicates the current server and current tree.

The following table describes the fields in NetWare Connections window:

Field	Description
Server Name	Names of the servers and trees the workstation is connected to.
User Name	Username for each connection.
Connection Number	User's connection number on the server.
Authentication State	Connections are either NDS or bindery connection.
NDS Tree	NDS Directory tree for each connection to a server that is running NetWare 4 or later.
Transport Type	The transit protocol in use between the server and the workstation.
Address	The internal address of the server.
Resource Type	Identifies the primary server.

Novell Client Information

The Novell Client window displays the information about the installed NetWare Client and its settings.

The following table describes the fields in the Novell Client window:

Field	Description
Preferred server	NetWare server that is used for NDS authentication of the user when the Novell Client for the Windows workstation software is started.
Preferred tree	Directory tree that the client first attaches to when the Novell Client for the Windows workstation software is started.
Name context	Current position or context in the NDS tree structure. This setting is applicable only to client workstations connecting to a NetWare 4 or NetWare 5 network.
First Network Drive	Network drive that is selected when you connect to a NetWare server.

Field	Description
Client Version	Novell Client 32 version number.

Network Protocols Information

The Network Protocols window displays the information about the active network protocols on a managed workstation using WinSock. The WinSock architecture also allows for simultaneous access to multiple transport protocols. WinSock contains the Windows Open System Architecture (WOSA) compliant architecture, which allows applications to access protocols including TCP/IP.

The following table describes the fields in the Network Protocols window:

Field	Description
Properties	Specifies characteristics of the protocol.
Address Family	Defines the structure of protocol addresses that are in use by the protocol.
Socket Type	Name represents the different socket types by the BSD socket interface. It can have the following values: ◆ Stream ◆ Datagram ◆ Raw Socket ◆ Seq. Packet ◆ RDM Socket ◆ Unknown
Protocol ID	Protocols identifier.

Field	Description
Message Size (Bytes)	Specifies the maximum message size (in bytes) supported by the protocol. This is the maximum size of a message that can be sent from or received by the host. For protocols that do not support message framing, the actual maximum size of a message that can be sent to a given address may be less than this value. If the protocol is stream-oriented, the concept of message size is not relevant. If the protocol is message-oriented, there is no maximum message size.
Protocol Name	Name of the protocol that is supported, such as TCP/IP, UDP/IP, or IPX.

Name Space Providers Information

The Name Space Providers window displays information about the Name Space Provider registered with WinSock Name Resolution and Registration APIs. WinSock 2 includes a new set of API functions that standardize how the applications access and use the various network naming services. This information will not be displayed for workstations with WinSock 1.1.

The following table describes the fields in the Name Space Providers Information window:

Field	Description
Name space	Specifies the name space (SAP, DNS, SLP).
Connected	Displays whether the Name Space Provider is enabled on the workstation.
Version	The name space version identifier.
Service Provider	Displays the string for the name space provider.

Network Drives Information

The Network Drives window displays information about mapped drives, drive capacity, volume label, file system information, sector size, and cluster size.

The following table describes the fields in the Network Drives window:

Field	Description
Drive Letter	Mapped drive letter.
Path	NetWare path of the volume or directory to which the drive is mapped. For example, if the directory ZENWORKS on SYS: volume of server ZEN_KYOTO is mapped to drive Q, the path displays ZEN_KYOTO\SYS:ZENWORKS.
File System	File system type for the mapped NetWare directory or volume.
Effective Rights	◆ Read For a folder, grants the right to open files in the folder and read the contents or run the programs. For a file, grants the right to open and read the file. ◆ Write For a folder, grants the right to open and change the contents of files in the folder. For a file, grants the right to open and write to the file. ◆ Create For a folder, grants the right to create new files and folders in the folder. For a file, grants the right to create a file and to salvage a file after it has been deleted. ◆ Delete Grants the right to delete the folder or file. ◆ Modify Grants the right to change the attributes or name of the folder or file, but does not grant the right to change its contents. Changing the contents requires the Write right. ◆ File Scan Grants the rights to see the folder or file with the DIR or NDIR command. ◆ Ownership Grants the ownership rights of the file, folder, or volume. If the corresponding rights are not given to the user, Effective Rights displays hyphen (-).

Field	Description
Long Name Size (Bytes)	Maximum length in characters of a filename component supported by the specified file system. For example, for a FAT file system supporting long names, the value is 255. The value for a DOS file system is 11.
Sector Size (Bytes)	Sector size in bytes.
Sectors Per Cluster	Number of sectors per cluster.
Total Clusters	Size of the volume in clusters.
Free Clusters	Number of clusters currently free for allocation. This number includes the space that is reclaimed from the sub-allocation file system and also clusters freed from deleted files.

Network Open Files Information

The Open Files window displays the names of open files on a file server. It lists the files opened in the network mapped drives, with the server name and connection ID.

The following table describes the fields in the Network Open Files window:

Field	Description
File Name	Name of the file.
Volume Name	Name of the Volume.
Server Name	Name of the file server.
User	The NetWare name under which the user's workstation is logged in to the file server.
Connection ID	Connection ID on which the file is opened.

Print Capture Information

The Print Capture window displays information about the captured queues, print options for each parallel port on the managed workstation, and current status of each port.

The following table describes the fields in the Print Capture window:

Field	Description
Printer Device Name	LPT device. Number of LPT ports for which captures can be managed.
Port State	Specifies whether the LPT device is captured.
Captured Queues	Captured print queue name.

VI Workstation Inventory

Inventory management involves collecting hardware and software inventory information from managed workstations. This inventory information is scanned and stored in a database, which the network administrator can access to manage the workstations.

ZENworks™ for Desktops (ZfD) lets you gather complete hardware and software inventory information in to a centralized database. The network administrator can query the inventory information of the managed workstations from ConsoleOne™.

The following sections will help you understand and use ZfD Workstation Inventory:

- "Understanding Workstation Inventory" on page 347
- "Setting Up Workstation Inventory" on page 449
- "Using Workstation Inventory" on page 511
- "Troubleshooting Workstation Inventory with Status Logs" on page 541

22 Understanding Workstation Inventory

The following sections describe the ZENworks™ for Desktops (ZfD) Workstation Inventory components and processes:

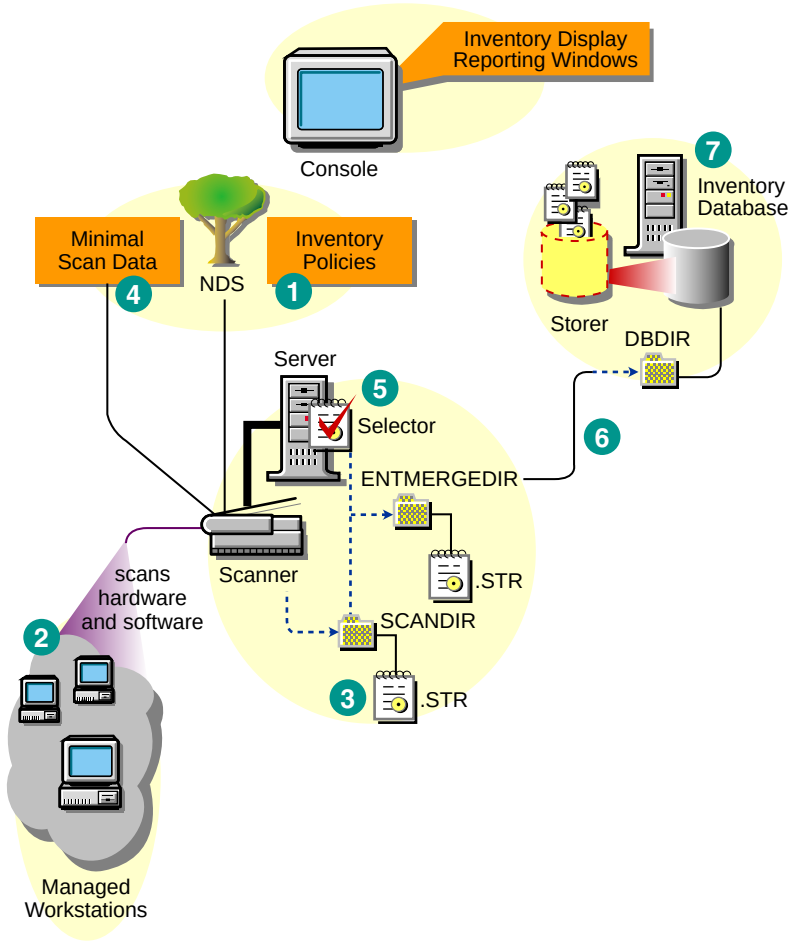
- "Understanding the Inventory Scanning Cycle" on page 347
- "Understanding Rolling Up Scan Data Across Servers" on page 349
- "Understanding the Inventory Scanner" on page 352
- "Understanding the Sender-Receiver" on page 366
- "Understanding the Selector" on page 375
- "Understanding the Storer" on page 377
- "Understanding the Service Manager" on page 379
- "Understanding the Effects of Workstation Inventory Installation" on page 386
- "List of Inventory Components on Servers" on page 390
- "Understanding the ZENworks for Desktops 3 Inventory Database Schema" on page 415

Understanding the Inventory Scanning Cycle

ZfD Workstation Inventory consists of the following components:

- Scanner
- Selector
- Sender
- Receiver
- Storer
- Inventory database

The following illustration depicts the scanning components and the scanning cycle, which is explained below:



The Scanners collect the workstation hardware and software information based on the Inventory policy settings for scheduling and collecting the scan data. This scan information is stored as scan data files (.STR) in the scan directories. A minimal subset of the scan data is also stored in the NDS® Workstation object.

The Selector on the server processes the scan data files to determine whether the new scan data should be merged with the existing data and then places the files in the appropriate directories.

If your inventory deployment requires roll up of data, the Sender and the Receiver on the servers transfer the compressed scan files from the lower-level servers to the higher-level servers. The Storer on the database server

stores the scan data files in the Inventory database. You view the workstation inventory information in ConsoleOne™.

The inventory scanning cycle is as follows:

1. The inventory policies in NDS define the inventory settings, such as scanning time, whether to include software scanning of workstations, and the location of the scan directory. These settings are customizable.
2. The Scanner reads the inventory policies and collects the workstation inventory information.
3. The Scanner stores the scan data of each workstation as a .STR file in the scan directory (SCANDIR) at the server.
4. The Scanner stores the minimal inventory information of the workstation in NDS.
5. The Selector validates the .STR file and places the file in the enterprise merge directory (ENTMERGEDIR). If there is a database attached, the Selector places the files in the Database directory (DBDIR).
6. The Storer updates the database with the inventory information of the .STR file.
7. The network administrator views the inventory information, queries the database, and generates inventory reports in ConsoleOne.

Understanding Rolling Up Scan Data Across Servers

If the inventory deployment rolls up scan data across servers, the process of scanning is as follows:

1. The inventory policies in NDS define the inventory settings, such as scanning time, whether to include software scanning of workstations, and the location of the scan directory. These settings are customizable.
2. The Scanner reads the inventory policies and collects the workstation inventory information.
3. The Scanner stores the scan data of each workstation as a scan data file (.STR) in the scan directory (SCANDIR) at the server.
4. The Scanner stores the minimal inventory information of the workstation in NDS.
5. The Selector validates the .STR file and places the file in the enterprise merge directory (ENTMERGEDIR) for roll up of scan data. If there is a

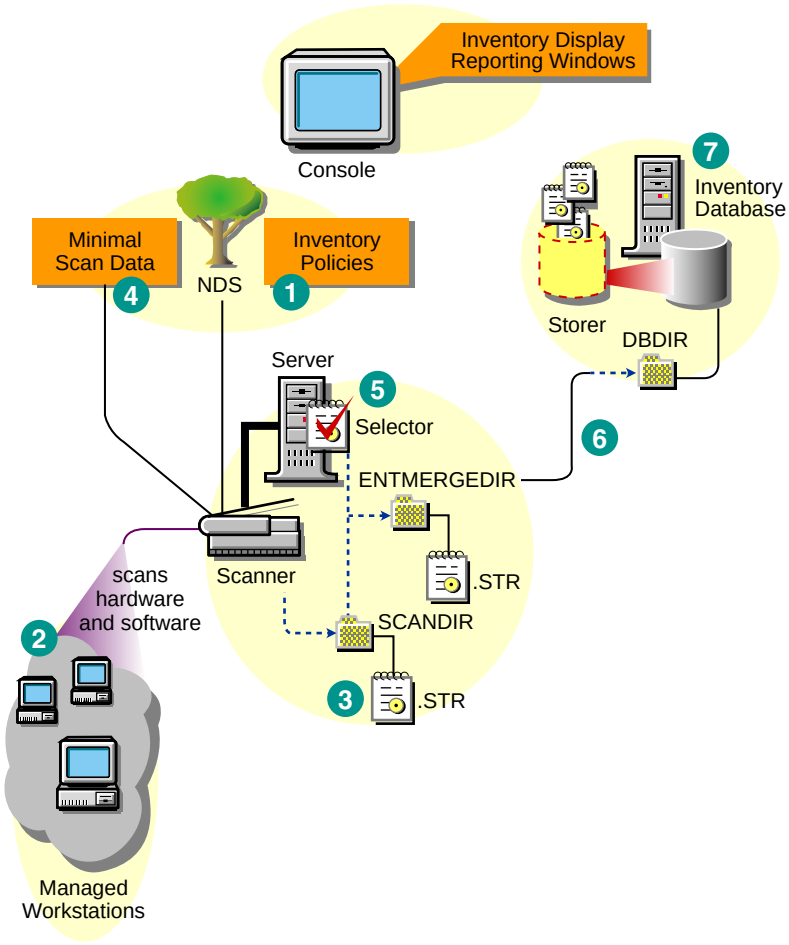
database attached, the Selector places the files in the database directory (DBDIR) also.

6. The Sender on the server has a Roll-Up policy to identify the server to which it will transmit the scan data and the Roll-Up Schedule specifies time for roll-up of data. The Sender compresses the .STR files as a .ZIP file and places the .ZIP file in the enterprise push directory (ENTPUSHDIR). The Sender then sends the .ZIP file to the Receiver on the next-level server.
7. The Receiver on the next-level server receives the .ZIP file.

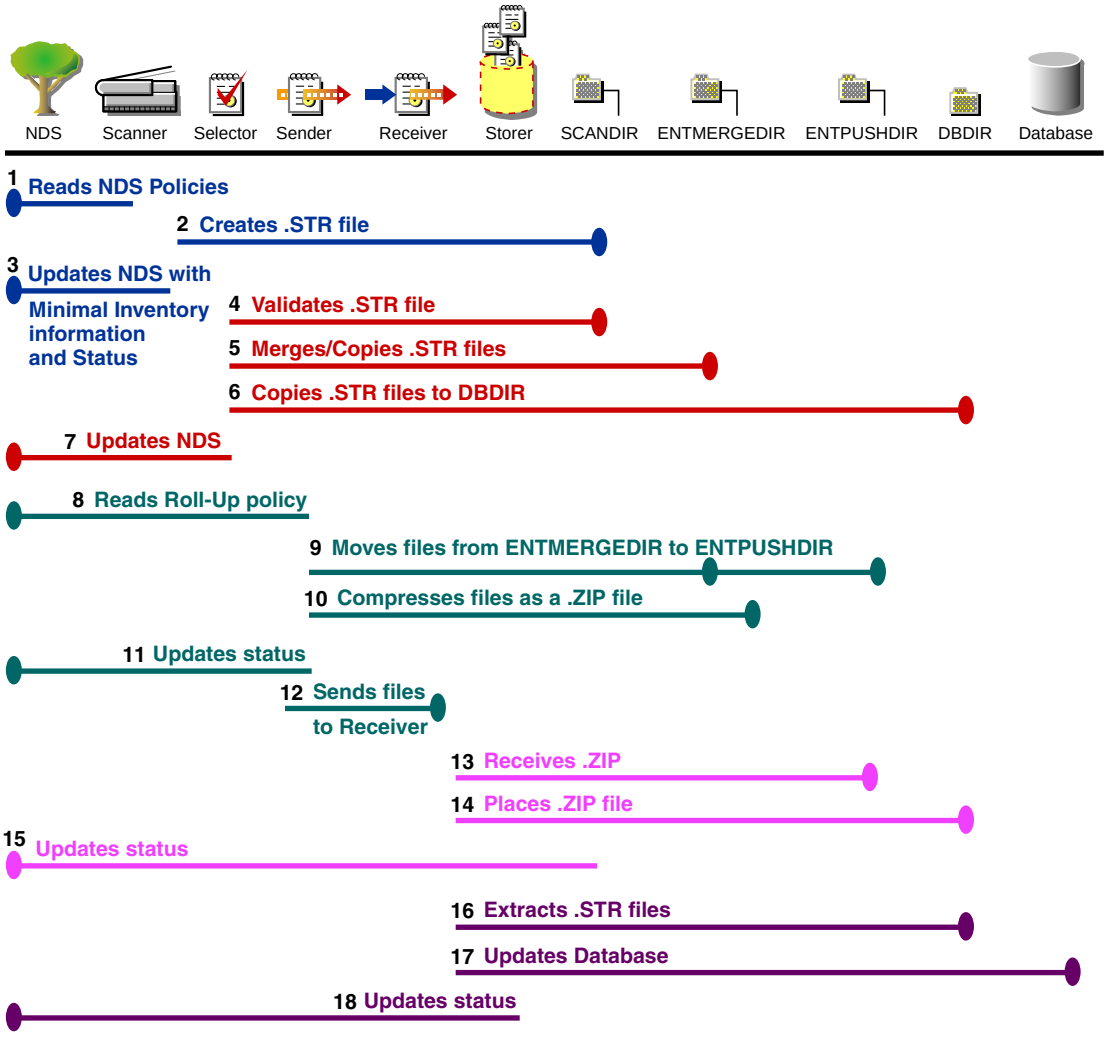
   On the Intermediate Server, the Receiver places the file in the enterprise push directory (ENTPUSHDIR). On the Intermediate Server with Database, or the Intermediate Server with Database and Workstations, the Receiver places the file in ENTPUSHDIR and copies the file in the Database Directory (DBDIR).

   On the Root Server, or the Root Server with Workstations, the Receiver places the file in DBDIR only.
8. The Storer extracts the .ZIP file containing the .STR files in to a temp directory (DBDIR\TEMP) and updates the database with the inventory information of the workstation .STR file.
9. The network administrator views the inventory information, queries the database, and generates inventory reports in ConsoleOne.

The following illustration depicts the scanning process across servers:



The sequence of scan operations is as follows:



Understanding the Inventory Scanner

ZfD uses the following platform-dependent scanners to collect workstation hardware and software information:

- WINSCAN.EXE to scan Windows* 95/98 workstations
- NTSCAN32.EXE to scan Windows NT* and Windows 2000 workstations

The scanners collect hardware details such as: floppy disk drive, hard disk drive, BIOS, bus, mouse, keyboard, display adapters, network adapter cards, modems, Jaz* drives, Zip* drives, sound cards, memory cards, serial ports, and parallel ports. The software scanning includes checking for applications on the workstations and reporting the information about the scanned software, such as the vendor name, and the product name and version.

The scan information collected by the scanners is stored as scan data files (.STR) in the scan directories (SCANDIR). A minimal set of scan data is also stored in the NDS Workstation object.

The following sections contain detailed information about the Inventory scanners:

- "How the Scanners Collect Workstation Inventory Data" on page 353
- "Scanning Process Flowchart" on page 355
- "Summary of Files that the Scanner Processes" on page 356
- "Software Information Collected by the Scanners" on page 357
- "DMI-Compliant Scanners" on page 357
- "Hardware Data Collected by the Scanners" on page 359

Also, see "Customizing the Software Scanning Information of Vendors and Products" on page 459.

How the Scanners Collect Workstation Inventory Data

The scanning process is as follows:

- The Workstation Inventory policy lets you configure the Scheduler to adjust scanning times at the workstations. You can set different actions to run the scanners on one or more managed workstations. See Understanding Scheduling of Policies and Policy Package for more information.
- The Scheduler triggers the scanner, which reads the following inventory settings from the Workstation Inventory policy and Inventory Service object:
  - **Software Scanning Option:** By default, the scanner collects the hardware information of the workstations. If the Enable Software Scan option is enabled in the Workstation Inventory policy, the scanner collects information about software applications.

- **Custom Scan Editor:** If the Software Scanning option is enabled, the scanner reports the software information of all .EXE files on the scanned workstations. You configure the applications that you want the scanner to collect information for using the Custom Scan Editor. See "Customizing the Software Scanning Information of Vendors and Products" on page 459 for more information.
- **Full Scan:** When scanning the workstation for the first time, the scanner collects the complete inventory of the workstation, referred to as a *full scan.* After the workstation is scanned, the next time the scanner compares the current inventory data to the history data that it maintains. If there are any changes to the workstation, the scanner reports the delta scan data, which contains only the changes in inventory since the last scan was reported. The Delta Scan setting is the default scan operation for each successive scan after the first scanning of the workstation.
- **Enable Scan of Workstation:** The scanner collects the inventory information of the workstations associated with the Inventory Service object when this option is enabled.
- **Location of the SCANDIR Path:** The scanner stores the scan information of the workstations in the scan data files (.STR) located in the scan directory (SCANDIR) on the inventory server.
- **Start Full Scan:** The scanner enforces a complete scanning of the workstation and this setting overrides the option set in the Inventory Service Object property page associated with the workstation.

  See "Setting Up Workstation Inventory" on page 449 for more information.

- The scanner collects the scan data based on the configurations of the inventory settings.

  If the workstation is instrumented for DMI, the scanner also queries the DMI Service Layer. See "DMI-Compliant Scanners" on page 357 for more information.

- The scan data of each workstation is stored as .STR files in the SCANDIR directory on the inventory server. The .STR file follows the filename convention: *macaddress_gmt_sequencenumber*.STR, where *macaddress* is the MacAddress of the workstation, *gmt* is the time at which the workstation is scanned for the first time, and *sequencenumber* is the internal sequencing number of the workstation. For example, 00508b12b2c4_944029836000_10.STR is the .STR file for the

workstation with the MACAddress of 00508b12b2c4, the GMT of 944029836000, and the internal sequencing number of 10.

- The scanner reports errors in the ZENERRORS.LOG file and updates the status information in the NDS Workstation object. The log file is stored in the WINDOWS\TEMP directory on Windows 95/98 workstations or in the TEMP directory on Windows NT/2000 workstations.

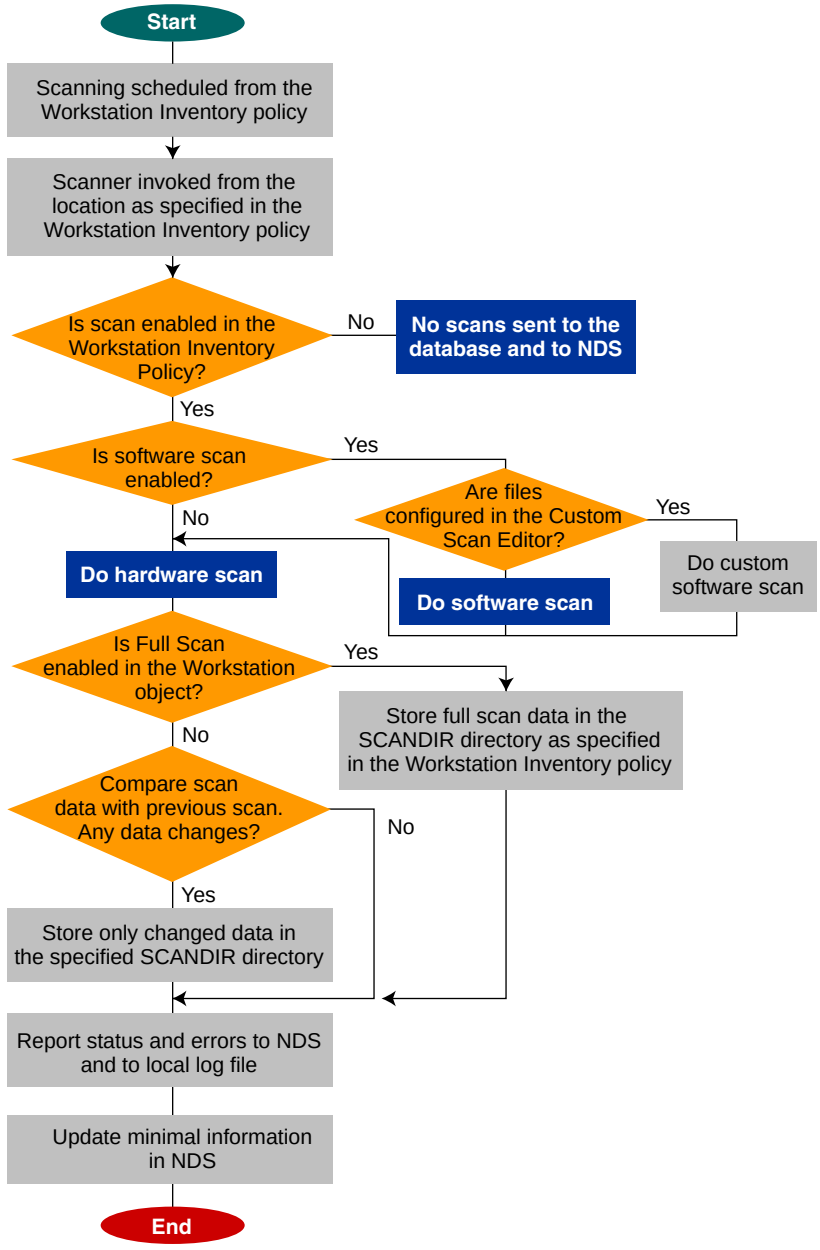
  The scanner tracks the changes in scan data by storing the scan data in the HIST.INI history file. For subsequent scans, the scanner compares the history scan data with the actual collected data and generates the .STR file.

- The scanner stores minimal information of the scan data in the MINFO.INI file located in WINDOWS\TEMP on Windows 95/98 workstations and in TEMP on Windows NT/2000 workstations.

  The scanner updates the NDS object with the scan data contained in MINFO.INI.

Scanning Process Flowchart

The following flowchart illustrates the hardware and software scanning process:


Start
Scanning scheduled from the Workstation Inventory policy
Scanner invoked from the location as specified in the Workstation Inventory policy
Is scan enabled in the Workstation Inventory Policy?
No
No scans sent to the database and to NDS
Yes
Is software scan enabled?
Yes
Are files configured in the Custom Scan Editor?
Yes
No
Do hardware scan
Do software scan
Do custom software scan
Is Full Scan enabled in the Workstation object?
Yes
Store full scan data in the SCANDIR directory as specified in the Workstation Inventory policy
No
Compare scan data with previous scan. Any data changes?
No
Yes
Store only changed data in the specified SCANDIR directory
Report status and errors to NDS and to local log file
Update minimal information in NDS
End

Summary of Files that the Scanner Processes

A summary of the files that the scanner processes follows:

Filename	Description	Location
filename.STR	Contains the scan data of each workstation.	Scan directory (SCANDIR).
HIST.INI	Contains the history of scan data for each workstation.	WINDOWS\TEMP on Windows 95/98 workstations, or TEMP on Windows NT/2000 workstations.
ZENERRORS.LOG	Contains the error and status information for the latest scan of the workstation.	WINDOWS\TEMP on Windows 95/98 workstations, or TEMP on Windows NT/2000 workstations.

Software Information Collected by the Scanners

The scanners follow this process for software scanning:

- Collects the information about the software on the workstations.
- Customizes the software scanning using the Custom Scan Editor.

  By default, the software scanning includes collecting information of files with .EXE file extensions. See "Customizing the Software Scanning Information of Vendors and Products" on page 459 for information on the list of software applications you can scan for.

  If the software applications on the workstation are installed using Microsoft* Installer, the scanners use the information from Microsoft Installer (MSI). Otherwise, the scanner collects the software information from the header of the software application files. See the Microsoft documentation for more information.
- Reports the information about the scanned software, such as name of the software product for each product version and the software vendor.

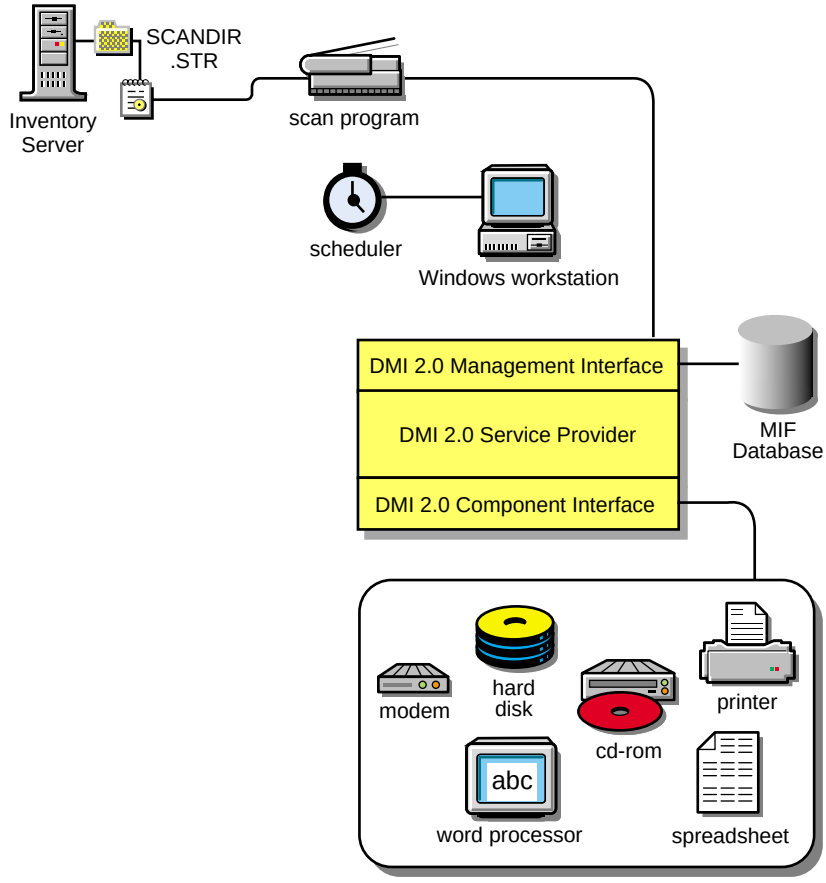
After the scan data is stored in the database, you can view, query, or generate reports of the software information.

DMI-Compliant Scanners

The scanners for scanning workstations (Windows 95/98 and Windows NT/2000) also include scanning based on the industry-standard Desktop Management Interface (DMI) specification 2.0. These programs use the Management Interface (MI) of DMI to look for the hardware components installed on the workstation. The scanners will scan for specific components that are instrumented on the workstation through DMI. The scanners will query the DMI service layer to retrieve this information.

The MI allows the DMI-compliant scanners to probe the Service Provider within the Service Layer. The Service Provider collects information from the manageable components and stores the collected information in the Management Information Format database. The Component Interface (CI) communicates with the manageable components and the Service layer.

The following figure shows the DMI-compliance architecture of the inventory scanners:



See the DMTF Web site (http://www.dmtf.org) for more information on DMI standards.

To scan the DMI data of the managed workstations, you need to instrument the workstation by installing the vendor-specific components. To install the DMI 2.0 Service Layer on the workstations, use the DMISLSnappshot.AOT file located in the \PUBLIC\ZENWORKS\DMISNAPSHOT directory to create a NAL Application Object.

HINT: If the workstations are DMI compliant, the scanners will collect hardware data by querying the DMI Service Layer. Otherwise, the scanners probe the workstations.

We recommend that you instrument DMI and also install DMI components that are supplied by the vendors.

For example, if you have a Compaq* Family Deskpro EN Model-SFF6500 workstation running under Windows 98, download the Management Product software - Compaq Insight Management Desktop Agents software for Windows 95/98/NT from the Compaq Web site.

For Dell* workstations, access the DM/Desktop Management Utilities software from the Dell Web site.

Hardware Data Collected by the Scanners

The Scanners collect the following hardware information. The list also contains the DMI components that are addressed in the DMI information.

Scan Data	DMI Class and Attribute
System.Type	Not applicable
System.MachineName	Not applicable
System.AssetTag	DMTF|System Enclosure|001.2
System.Model	Not applicable
System.ModelNumber	Not applicable
System.DNName	Not applicable
System.TreeName	Not applicable
IP.Address	Not applicable
IP.Subnet (Subnet Mask)	Not applicable
IP.HostName	Not applicable
NetworkAdapter.MACAddress	Not applicable
Model.Description	Not applicable
Model.Name	Not applicable
Model.Vendor	Not applicable
NetworkAdapter.DriverDescription	DMTF|Network Adapter Driver |001.Driver Software Name
NetworkAdapter.DriverName	DMTF|Network Adapter Driver |001.Driver Software Name

Scan Data	DMI Class and Attribute
NetworkAdapter.DriverVersion	DMTF|Network Adapter Driver |001.Driver Software Version
NetworkAdapter.Speed	DMTF|Network Adapter 802 Port|001.5
NetworkAdapter.Name	DMTF|Network Adapter Driver|001.Driver Software Description
NetworkAdapter.PermAddress	DMTF|Network Adapter 802 Port|001.2
Login.LoginName	Not applicable
Login.DomainName	Not applicable
NWClient.Version	Not applicable
Processor.stepping	DMTF|Processor|004.4
Processor.DeviceID	Not applicable
Processor.Family	DMTF|Processor|004.3
Processor.MaxClockSpeed	DMTF|Processor|004.5
Processor.Role	DMTF|Processor|004.2
Processor.Upgrade	DMTF|Processor|004.7
Processor.OtherFamily	Not applicable
BIOS.Manufacturer	Not applicable
BIOS.BIOSDate	Not applicable
BIOS.BIOSIDBytes	Not applicable
BIOS.Copyright	Not applicable
BIOS.SerialNumber	Not applicable
BIOS.BIOSType	Not applicable
BIOS.PrimaryBIOS	Not applicable
BIOS.Size	Not applicable
Bus.Version	Not applicable

Scan Data	DMI Class and Attribute
Bus.Type	DMTF|Bus Port|001.Protocol
Mouse.DeviceType	Not applicable
Mouse.Type	DMTF|Mouse|004.Mouse Interface
Mouse.NumberOfButtons	DMTF|Mouse|004.Mouse Buttons
Mouse.DriverName	DMTF|Mouse|004.Mouse Driver Name
Mouse.DriverVersion	DMTF|Mouse|004.Mouse Driver Version
Mouse.IRQ	DMTF|Mouse|004.Mouse IRQ
Keyboard.Layout	DMTF|Keyboard|003.Layout
Keyboard.Subtype	Not applicable
Keyboard.Type	DMTF|Keyboard|003.Keyboard.Type
Keyboard.Fkeys	Not applicable
Keyboard.Delay	Not applicable
Keyboard.TypematicRate	Not applicable
Monitor.NumberOfColorPlanes (NEW)	Not applicable
Monitor.HorizontalResolution	DMTF|Video|004.Current Horizontal Resolution
Monitor.VerticalResolution	DMTF|Video|004.Current Vertical Resolution
Monitor.DisplayType	DMTF|Video|004.Video Type
Monitor.MemoryType	DMTF|Video|004.Video Memory Type
Monitor.MaxMemorySupported	DMTF|Video|004.Video RAM Memory Size
Monitor.Bitsperpixel	DMTF|Video|004.Current Number of Bits per Pixel
Monitor.ControllerDescription	DMTF|Video|004.Video Controller Description
Monitor.MaxRefreshrate	DMTF|Video|004.Maximum Refresh Rate
Monitor.MinRefreshrate	DMTF|Video|004.Minimum Refresh Rate
Monitor.VideoBIOSManufacturer	DMTF|Video BIOS|001.BIOS Manufacturer

Scan Data	DMI Class and Attribute
Monitor.VideoBIOSVersion	DMTF|Video BIOS|001.Video.BIOS Version
Monitor.VideoBIOSReleaseDate	DMTF|Video BIOS|001.Video.BIOS Release Date
Monitor.VideoBIOS.IsShadowed	DMTF|Video BIOS|001.Video.Shadowing State
ParallelPort.Name	DMTF|Parallel Ports|003.Logical Name
ParallelPort.DMASupport	DMTF|Parallel Ports|003.DMA Support
ParallelPort.IRQ	DMTF|Parallel Ports|003.IRQ Used
SerialPort.Name	DMTF|Serial Ports|004.Logical Name Pointing Device Port Name
SerialPort.Address	DMTF|Serial Ports|004.Serial Base I/O Address
SerialPort.IRQ	DMTF|Serial Ports|004.IRQ Used
FloppyDrive.Dummy	Not applicable
FloppyDrive.DriverName	DMTF|Logical Drives|001.Logical Drive Name (when DMTF|Logical Drives|001.Logical Drive Type=Floppy Drive(7))
FloppyDrive.Vendor	Not applicable
FloppyDrive.Description	DMTF|Disks|003.Interface Description (when DMTF|Disks|003.Storage Type=Floppy Disk(4))
FloppyDrive.MaxNumberofCylinders	DMTF|Disks|003.Number of Physical Cylinders
FloppyDrive.NumberOfHeads	DMTF|Disks|003.Number of Physical Heads
FloppyDrive.Sectors	DMTF|Disks|003.Number of Physical Sectors Per Track
FloppyDrive.Size	DMTF|Disks|003.Total Physical Size
CIM_CDROMDrive.Dummy	Not applicable
CDROMDrive.Name	DMTF|Logical Drives|001.Logical Drive Name
CDROMDrive.Vendor	Not applicable

Scan Data	DMI Class and Attribute
CDROMDrive.Description	DMTF|Disks|003.Interface.Description (when DMTF|Disks|003.Storage Type=Compact Disk Disk(8))
CDROMDrive.DeviceID	DMTF|Disks|003.DeviceID
HardDrive.Dummy	Not applicable
HardDrive.Vendor	Not applicable
HardDisk.Description	DMTF|Disks|003.Interface Description (when DMTF|Disks|003.Storage Type=Hard Disk(3))
HardDisk.Cylinders	DMTF|Disks|003.Number of Physical Cylinders
HardDisk.Heads	DMTF|Disks|003.Number of Physical Heads
HardDisk.Sectors	DMTF|Disks|003.Number of Physical Sectors per Track
HardDisk.Capacity	DMTF|Disks|003.Total Physical Size
Storage.Type	DMTF|Disk|003.Storage Type
HardDisk.PartitionName	DMTF|Partition|002.Partition Name
HardDisk.PartitionSize	DMTF|Partition|002.Partition Size
HardDisk.PartitionFileSystemType	DMTF|Partition|002.FileSystem
LogicalDrive.Name	DMTF|Logical Drives|001.Logical Drive Name (when DMTF|Logical Drives|001.Logical Drive Type=Fixed Drive(3))
LogicalDrive.Size	DMTF|Logical Drives|001.Logical Drive Size
LogicalDrive.VolumeSerialNumber	Not applicable
LogicalDrive.Volume (Volume Label)	Not applicable
LogicalDrive.Type	DMTF|Logical Drives|001.Logical Drive Type
Operating System.Name	DMTF|Operating System|001.Operating System Name
OperatingSystem.Version	DMTF|Operating System|001.Operating System Version
OperatingSystem.Codepage	Not applicable

Scan Data	DMI Class and Attribute
OperatingSystem.InstallDate	Not applicable
OperatingSystem.TotalSwapSpaceSize	DMTF|System Memory Settings|001.Total Size of Paging Files
OperatingSystem.Description	DMTF|Operating System|001.Operating System Description
VirtualMemory.TotalVirtualMemorySize	DMTF|System Memory Setting|001.Total Virtual Memory
InventoryScanner.Version	Not applicable
InventoryScanner.LastScanDate	Not applicable
InventoryScanner.InventoryServer	Not applicable
SoundCard.Description	Not applicable
SoundCard.Name	Not applicable
SoundCard.Manufacturer	Not applicable
Scsidrive.Description	Not applicable
Scsidrive.Manufacturer	Not applicable
Scsidrive.Name (mapped drive)	Not applicable
Memory.Size	DMTF|System Memory Settings|001.Total Physical Memory
Cache.Level	DMTF|System Cache|003.System Cache Level
Cache.WritePolicy	DMTF|System Cache|003.System Cache Write Policy
Cache.ErrorCorrection	DMTF|System Cache|003.System Cache Error Correction
Cache.Type	DMTF|System Cache|003.System Cache Type
Cache.LineSize	DMTF|System Cache|003.Line Size
Cache.ReplacementPolicy	DMTF|System Cache|003.Replacement Policy
Cache.ReadPolicy	DMTF|System Cache|003.Read Policy
Cache.Associativity	DMTF|System Cache|003.Associativity
Cache.Speed	DMTF|System Cache|003.System Cache Speed

Scan Data	DMI Class and Attribute
Cache.Size	DMTF|System Cache|003.System Cache Size
Motherboard.Verify	Not applicable
MotherBoard.BoardRev	Not applicable
MotherBoard.Slots	DMTF|Motherboard|001.Number of Expansion slots
Battery.Name	DMTF|Portable Battery|002.Portable Battery Device Name
Battery.Chemistry	DMTF|Portable Battery|002.Portable Battery Device Chemistry
Battery.Capacity	DMTF|Portable Battery|002.Portable Battery Design Capacity
Battery.Voltage	DMTF|Portable Battery|002.Portable Battery Design Voltage
Battery.Version	DMTF|Portable Battery|002.Portable Smart Battery Version
Battery.Manufacturer	DMTF|Portable Battery|002.Portable Battery Manufacturer
Battery.ManufacturerDate	DMTF|Portable Battery|002.Portable Battery Manufacturer Date
Battery.SerialNumber	DMTF|Portable Battery|002.Portable Battery Serial Number
PowerSupply.InputVoltageDescription	DMTF|Power Supply|002.Power Suppy Input Voltage Capability Description
PowerSupply.Power	DMTF|Power Supply|002./Total Output Power
DMA.Number	DMTF|DMA|001.DMA Number
DMA.Description	DMTF|DMA|001.DMA Description
DMA.Availability	DMTF|DMA|001.DMA Channel Availability
DMA_BurstMode	DMTF|DMA|001.DMA BurstMode
UCS.PrimaryOwnerContact	DMTF|General Information|001.3
UCS.PrimaryOwnerName	DMTF|General Information|001.4
IRQ.Number	DMTF|IRQ|002.IRQNumber
IRQ.Availability	DMTF|IRQ|002.Availability

Scan Data	DMI Class and Attribute
IRQ.TriggerType	DMTF|IRQ|002.TriggerType
IRQ.Shareable	DMTF|IRQ|002.Shareable
CIM_Card.Description	DMTF|System Slots|003.Description
CIM_Slot.MaxDataWidth	DMTF|System Slots|003.MaxDataWidth
CIM_Slot.ThermalRating	DMTF|System Slots|003.Tr

HINT: PCMCIA modems are connected to the computer through the PCMCIA slots on the workstations. The Scanner detects PCMCIA modems that are active on the computer. If you want to know which modem is installed on the computer, use the Windows System Device Manager on the Windows workstation.

Non-PCMCIA modems are connected to the computer through the external ports. For example, some non-PCMCIA modems are connected through the serial ports. The Scanner detects Non-PCMCIA modems that are installed on the computer.

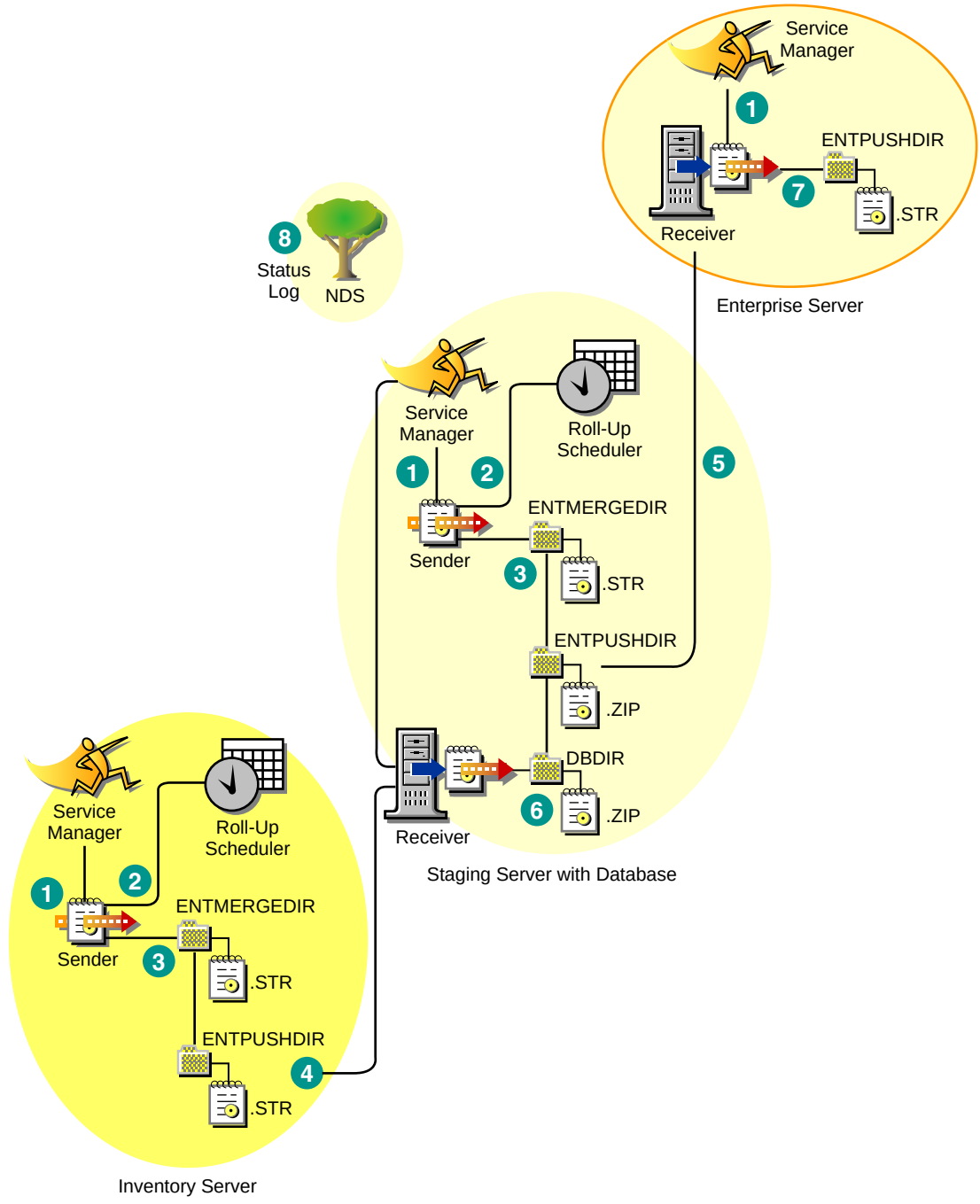
Non-PCMCIA modems may not be active at the time of scanning. Also, these modems may not be connected, though configured on the computer. In this case, the Scanner detects the modem and reports the scan information of the modem.

Understanding the Sender-Receiver

The Sender and the Receiver on the servers transfer the scan files from the lower-level servers to the higher-level servers. The following sections contain more information:

- “Understanding the Sender” on page 369
- “Understanding the Receiver” on page 370
- “Understanding the Compressed Scan Data File” on page 371
- “Quick Reference of Sender-Receiver” on page 371

The following illustration depicts the processing done by the Sender-Receiver.


Service Manager
1
ENTPUSHDIR
7
.STR
Receiver
Enterprise Server
8
Status Log
NDS
Service Manager
Roll-Up Scheduler
1
2
ENTMERGEDIR
Sender
3
.STR
ENTPUSHDIR
.ZIP
5
DBDIR
6
.ZIP
Receiver
Staging Server with Database
Service Manager
Roll-Up Scheduler
1
2
ENTMERGEDIR
Sender
3
.STR
ENTPUSHDIR
4
.STR
Inventory Server

The processing done by the Sender-Receiver is as follows:

1. The Service Manager starts the Sender-Receiver component.
2. The Roll-Up Scheduler activates the Sender at the specified roll-up time.
3. The Sender moves the scan data files (.STR) from the enterprise merge directory (ENTMERGEDIR) to the enterprise push directory (ENTPUSHDIR) and compresses the files as a .ZIP file.
4. The Sender sends the .ZIP file from the ENTPUSHDIR directory to the Receiver on the next-level server.
5. The Sender on the Intermediate Server sends the .ZIP file to the Receiver on the next-level server.
6. The Receiver copies the .ZIP files to the ENTPUSHDIR directory.
7. The Receiver copies the .ZIP files to the database directory (DBDIR).
8. The Sender-Receiver logs the status in NDS.

Understanding the Sender

The Sender is a Java* component that runs on any Leaf Server or on the Intermediate Server. The Sender is a service loaded by the Service Manager. See "List of Inventory Components on Servers" on page 390 for a quick reference table of server components.

The flow of information from the Sender in the roll-up of scan data is as follows:

1. The Service Manager starts the Sender on the server. At the specified time scheduled in the Roll-Up Schedule, the Sender moves the scan data files (.STR) from the enterprise merge directory (ENTMERGEDIR) to the enterprise push directory (ENTPUSHDIR).

   The Sender compresses these .STR files in the ENTPUSHDIR directory of the server as a .ZIP file and then deletes the .STR files. See "Understanding the Compressed Scan Data File" on page 371 for more information.

2. The Sender creates a new record in the zeninvRollUpLog attribute of the Inventory Service object (ZenInvservice) in NDS with the following details: server on which the Sender compresses the STR files and the name and size of the .ZIP file.

3. Based on the Discard Scan Data Time in the Inventory Service object properties of the Receiver, the Sender deletes the compressed .ZIP files in the ENTPUSHDIR directory that have been created earlier than the specified discard scan data time. This removes unwanted scan information being sent in the roll-up.
4. The Sender sends the compressed .ZIP files to the Receiver, with the oldest compressed files sent first.
5. The Sender receives an acknowledgment from the Receiver that a .ZIP file was properly received and then deletes the compressed files in the ENTPUSHDIR directory.
6. After the roll up of data, the Sender updates the zeninvRollUpLog attribute of the server on which the compressed file was created with the following details: server from which the Sender transmitted the file, name of the .ZIP file, time of transmission, total time taken to transmit the files, and the server to which it was sent.

   The status information for all actions of the Sender is logged in the Roll-Up Log and Server Status log. See "Troubleshooting Workstation Inventory with Status Logs" on page 541 for more information.

If the Sender is unable to connect to the Receiver, the Sender retries to connect after 10 seconds. The time interval increases exponentially by a factor of 2. After 14 retries, the Sender stops trying to connect to the Receiver. The Sender retries for approximately 23 hours before it discontinues trying. The Sender does not process any other data while it is establishing the connection.

Understanding the Receiver

The Receiver is a Java component that runs on the Intermediate Server or on the Root Server. The Receiver is a service loaded by the Service Manager. See "List of Inventory Components on Servers" on page 390 for a quick reference table of server components.

The processing done by the Receiver is as follows:

1. On successfully establishing communication with the Sender, the Receiver receives the scan .ZIP file from the Sender. The file is placed in the enterprise push directory (ENTPUSHDIR).

   On an Intermediate Server, the file is placed in ENTPUSHDIR. On an Intermediate Server with Database, or an Intermediate Server with Database and Workstations, the file is placed in ENTPUSHDIR and copied to the Database Directory (DBDIR).

2. The Receiver on the Root Server or the Root Server with Workstations receives the .ZIP files from the Senders and copies the files to the DBDIR directory on the server.
3. The Receiver logs the status information in the Roll-Up log. See "Troubleshooting Workstation Inventory with Status Logs" on page 541 for more information.

Understanding the Compressed Scan Data File

The Sender compresses the scan data files (.STR) into a .ZIP file. The .ZIP file is named using the following naming conventions:

scheduledtime_inventoryservername_siteID_sitename.ZIP

where, *scheduledtime* refers to the date and time when the Sender is scheduled for roll-up of scan information, *inventoryservername* refers to the inventory server on which the .ZIP file was compressed, *siteID* refers to the identification of the database that is attached to the inventory server, *sitename* refers to the unique Site name of the database specified during installation, and *ZIP* is the file extension for the compressed files.

The .ZIP filename changes depending on if the database is attached to the server. If the database is not attached to the server, the file is named as follows:

*scheduledtime_inventoryservername*_NOTSTOREDINDATABASETILLNOW_NULL.ZIP

The .ZIP file contains the .STR files and a property file. The property file is named using the following conventions:

scheduled_time_inventoryservername.PRP

The property file identifies the information for roll-up from the enterprise push directory (ENTPUSHDIR) to the next-level server. The property file contains the scheduled time, inventory server name, and signature. The signature helps to authenticate the .ZIP file.

Each .ZIP file can contain a maximum of 1,000 .STR files.

Quick Reference of Sender-Receiver

The following table provides a quick reference of the directories that the Sender-Receiver uses:

On the Leaf Server, and the Leaf Server with Database:

Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
✓	--	◆ Sender moves the STR files to the ENTPUSHDIR.	◆ Sender compresses the .STR files as a .ZIP file. ◆ Sender deletes the .STR files. ◆ Sends the .ZIP file to the next-level server.	--

On the Intermediate Server:

Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
✓	✓	--	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender sends the .ZIP files to the next-level server.	--

On the Intermediate Server with Workstations:

Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
✓	✓	◆ Sender moves the .STR files to the ENTPUSHDIR.	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender compresses the .STR files in to .ZIP files. ◆ Sender deletes the .STR files. ◆ Sender sends the .ZIP files to the next-level server.	

On the Intermediate Server with Database:

Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
✓	✓	--	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender sends the .ZIP file to the next-level server.	◆ Receiver copies the file in this directory.

On the Intermediate Server with Database and Workstations:

Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
✓	✓	◆ Sender moves the .STR files to the ENTPUSH DIR.	◆ Receiver receives the .ZIP files from the lower-level server in this directory. ◆ Sender compresses the .STR files as a .ZIP file. ◆ Sender deletes the .STR files. ◆ Sender sends the .ZIP file to the next-level server.	◆ Receiver copies the file in this directory.

On the Root Server, and Root Server with Workstations:

Sender	Receiver	ENTMERGDIR	ENTPUSHDIR	DBDIR
--	✓	--	--	◆ Receiver receives the .ZIP files from the lower-level server in this directory.

On the Standalone Server, the Receiver is not loaded.

Understanding the Selector

The Selector is a Java component on the server that receives the scan data from the workstations. These servers can be any of the following: Leaf Server, Leaf Server with Database, Intermediate Server with Database and Workstations, Intermediate Server with Workstations, Root Server with Workstations, and Standalone Server. See "List of Inventory Components on Servers" on page 390 for a quick reference table of server components.

The processing done by the Selector is as follows:

1. While scanning the workstation, the Scanner creates a scan data file (.STR) in the scan directory (SCANDIR) at the server for each scan done on the workstation. The location of SCANDIR is obtained from the Inventory Service object. The Selector processes the .STR files placed by the Scanner in the SCANDIR directory.
2. The Selector checks the validity of the .STR file to ensure that it is a valid file generated by the Scanner. The Selector processes only valid .STR files. If invalid files are present in the directory, the Selector deletes them.
3. Based on the role of the server, the Selector copies the .STR files to the appropriate directories:
   - If a full scan is done at the workstation, the Selector on the server copies the .STR file to the DBDIR directory (if the database is attached) and the ENTMERGE directory. If the .STR file already exists in the directory, it overwrites the file. See the table below.
   - If a delta scan is done at the workstation, the Selector on the server appends the contents of the file in to the existing files in the DBDIR directory (if the database is attached) and the ENTMERGE directory.

   The following table lists the directories that the Selector copies the files to:

Server	Copies the .STR file to the Database Directory (DBDIR)	Copies the .STR file to the Enterprise Merge Directory (EntMergeDir)
Leaf Server with Database	✓	✓
Leaf Server	--	✓

Server	Copies the .STR file to the Database Directory (DBDIR)	Copies the .STR file to the Enterprise Merge Directory (EntMergeDir)
Intermediate Server with Database & Workstations	✓	✓
Standalone Server	✓	--
Root Server with Workstations	✓	--

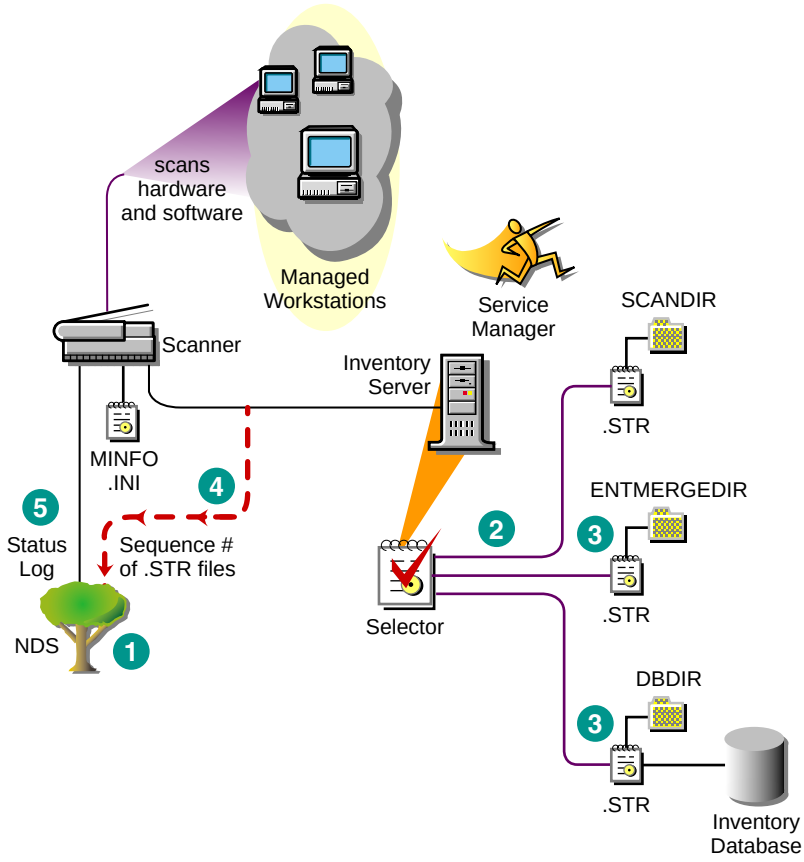
4. The Selector updates the sequence number of valid .STR files in NDS.

   The Selector determines whether to enforce a full scan if the scanning sequence number in the .STR file is improper. If the .STR file is invalid or if there are discrepancies in the sequence number of the .STR file, the Selector enforces a full scan.

5. The Selector logs the status in the Server log. See "Troubleshooting Workstation Inventory with Status Logs" on page 541 for more information.

   The Selector removes the existing .STR files in the SCANDIR directory.

The processing done by the Selector is as follows:



Understanding the Storer

The Storer is a Java component on the server that has a database attached to it. These servers can be any of the following: Leaf Server with Database, Intermediate Server with Database, Intermediate Server with Database and Workstations, Root Server, and Root Server with Workstations. See “List of Inventory Components on Servers” on page 390 for a quick reference table of server components.

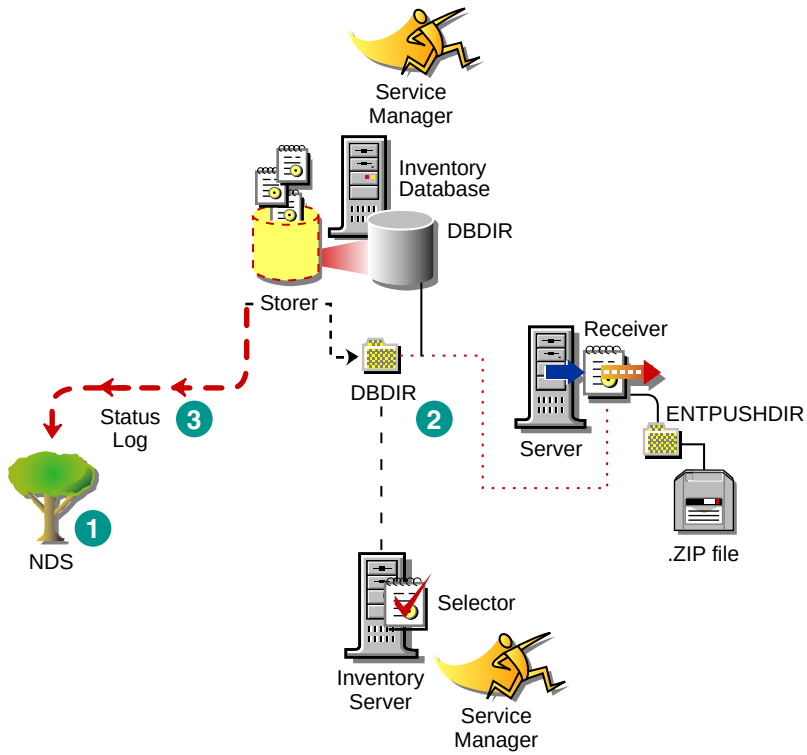
The Storer processes the files in the DBDIR directory. The Storer runs as a Service loaded by the Service Manager.

The processing done by the Storer is as follows:

1. From the server configuration information stored in NDS, the Storer looks in the database directory (DBDIR) for the scan files. The server configuration information determines the location of DBDIR and the database server from the NDS policy. The Selector places the .STR files in DBDIR and the Receiver places the scan .ZIP files in DBDIR.
2. The Storer extracts the .ZIP file containing the .STR files to a temp directory (DBDIR\TEMP) and updates the database with the inventory information of the .STR file for a workstation.

   The Storer forces a full scan of a workstation if there are problems with the .STR files.
3. The Storer updates the status in the Workstation Status log and updates the Roll-Up log.You can view the status information in the Server Status log. See "Troubleshooting Workstation Inventory with Status Logs" on page 541 for more information.

The following illustration depicts the processing done by the Storer:



Understanding the Service Manager

The Inventory Service Manager loads the inventory components on the server, based on the configuration parameters specified in the server properties file.

This sections contains the following:

- "List of Services" on page 379
- "Services on NetWare Servers" on page 383
- "Services on Windows NT/2000 Servers" on page 384

List of Services

The Service Manager loads the following services:

Service Name	Description
Server Configuration Service	Loads the server configuration services
Inventory Scheduler Service	Loads the Inventory Scheduler
Selector Service	Loads the Selector
Receiver Service	Loads the Receiver
Sender Service	Loads the Sender
Storer Service	Loads the Storer
NOTE: Support Pack 1	**NOTE: Support Pack 1**
Inventory Upgrade Service	Runs the upgrade utilities for inventory
NDSLookupForDB Service	Loads the service for NDSLookupForDB tool
DBDelete Service	Loads the service for DBDelete tool
DBBackup Service	Loads the service for DBBackup tool

The AlterDBSpace tool loads the AlterDBSpace Service and the ZEN2 Delete tool uses the zen2remove Service.

Property File: There are property files that load the different services on the server depending on the role of the server. The name of the property file indicates the role of the server. Only the required services are loaded as per the role of the server. The property files should not be modified.

A sample role based property file for Leaf Server with Database is as follows:

```
[Server Configuration Service]

type = system

Load Sequence = 0

Load Option = auto

Class Name =
com.novell.zenworks.desktop.inventory.servercommon.ServerCon
fig

Arguments =
```

NOTE: Support Pack 1

```
[Upgrade Service]

type = user

Load Sequence = 1

Load Option = auto

Class Name =
com.novell.zenworks.desktop.inventory.upgradeService.Upgrade
Service

Arguments =

[Inventory Scheduler Service]

type = system

Load Sequence = 2

Load Option = auto

Class Name =
com.novell.zenworks.desktop.inventory.servercommon.InventorS
cheduler

Arguments =

[Selector Service]
```

```
type = user

Load Sequence = 3

Load Option = auto

Class Name =
com.novell.zenworks.desktop.inventory.selector.SelectorServi
ceInit

Arguments =

[Storer Service]

type = user

Load Sequence = 4

Load Option = auto

Class Name =
com.novell.zenworks.desktop.inventory.storer.StorerServiceIn
it

Arguments =

[Sender Service]

type = user

Load Sequence = 5

Load Option = auto

Class Name =
com.novell.zenworks.desktop.inventory.senderreceiver.control
.SenderServiceInit

Arguments =

[NDSLookupForDB Service]

type = user

Load Sequence = 6

Load Option = manual

Class Name =
com.novell.zenworks.desktop.inventory.dbutilities.NDSLookupF
orDB

Arguments = "WSDELETE.LOK"
```

```
[DBDelete Service]

type = user

Load Sequence = 7

Load Option = manual

Class Name =
com.novell.zenworks.desktop.inventory.dbutilities.DBDelete

Arguments = "WSDELETE.LOK"

[DBBackup Service]

type = user

Load Sequence = 8

Load Option = manual

Class Name =
com.novell.zenworks.desktop.inventory.dbutilities.DBBackup

Arguments = "Backup"
```

Do not modify these property files as services or the Service Manager cannot be loaded.

Depending on the role of the server, the server properties files include:

Server Type	Server Property File
Root Server	ROOT_DB.PROPERTIES
Root Server with Workstations	ROOT_DB_WKS.PROPERTIES
Intermediate Server	INT.PROPERTIES
Intermediate Server with Workstations	INT_WKS.PROPERTIES
Intermediate Server with Database	INT_DB.PROPERTIES
Intermediate Server with Database and Workstations	INT_DB_WKS.PROPERTIES
Leaf Server	LEAF_WKS.PROPERTIES
Leaf Server with Database	LEAF_DB_WKS.PROPERTIES
Standalone Server	STANDALONE.PROPERTIES

The Inventory Service Manager reads the server properties file (CONFIG.PROPERTIES) and the role based property file in the \PUBLIC\ZENWORKS\WMINV\PROPERTIES directory, and loads the required services and server components.

The contents of the CONFIG.PROPERTIES file are as follows:

```
NDSTREE=treename

INVENTORYSERVICEDN=dn_of_the_inventory_service_object

SINGLETONPORT=65433
```

Services on NetWare Servers

On a NetWare server, the installation program modifies the ZFDSTART.NCF file located in SYS:\SYSTEM directory to load STARTINV.NCF. The STARTINV.NCF file located in the SYS:\SYSTEM brings up the Inventory Service Manager at server startup time.

On a NetWare sever, the installation program adds the STARTINV.NCF entry in the ZFDSTART.NCF file.

The contents of the ZFDSTART.NCF file are as follows:

```
;ZENWORKS for Desktop 3 Settings

STARTINV.NCF
```

The contents of the STARTINV.NCF file are as follows:

```
InvEnv

Naming

java -
DConfigFile=sys:\public\zenworks\wminv\properties\Config.pro
perties -nsac -jszenWSInv -neh -mx128m -classpath
$tmppath;$classpath
com.novell.zenworks.desktop.inventory.servercommon.ZENWorksI
nventoryServiceManager
```

You can start, stop, or list the services, if the Inventory Service Manager is already loaded.

To check if the Inventory Service Manager is loaded, at the server prompt, enter **`java -show`**.

This will display the following message:

```
com.novell.zenworks.inventory.servercommon.ZENWorksInventory
ServiceManager
```

To start a service:

1. At the server prompt, enter **StartSer *service_name***

   service_name refers to any of the listed services. Follow the service naming syntax when you modify the *service_name*.

   For example, to start the Storer, enter **StartSer *Storer***

To stop a service:

1. At the server prompt, enter **StopSer *service_name***

   service_name refers to any of the listed services. Follow the service naming syntax when you modify the *service_name*.

   For example, to start the Storer, enter **StopSer *Storer***

   To stop all services, enter **StopSer ***

To list a service:

1. Edit the LISTSER.NCF file located in the SYS:\SYSTEM directory.
2. Modify the *service_name* in the following line:

   ```
   java -neh - classpath $tmppath;CLASSPATH
   com.novell.zenworks.desktop.inventory.servercommon.ZENWo
   rksInventoryListService "service_name"
   ```

service_name specified in the .NCF file refers to any of the listed services. Follow the service naming syntax when you modify the *service_name*.

For example, to list Selector Server, the *service_name* specified in the file should be "Selector Service".

To refer to all services, use the asterisk (*) wildcard character within double quotes "*". This wildcard character can be used with Stop and List services parameters.

Services on Windows NT/2000 Servers

On Windows NT/2000 servers, the installation program creates the Service Manager as a service. During server startup, this Inventory Service Manager is loaded as a service.

You can start, stop, or list the services, if the Inventory Service Manager (ZENworks Inventory Service) is already loaded.

To start a service:

1. Go to the PUBLIC\ZENWORKS\WMINV\BIN directory.
2. At the prompt, enter **`StartSer "service_name"`**.

   where *service_name* refers to an Inventory service.

To stop a service:

1. Go to the PUBLIC\ZENWORKS\WMINV\BIN directory.
2. At the prompt, enter **`StopSer"service_name"`**.

   where *service_name* refers to an Inventory service.

To stop all services (ZENworks Inventory Service), use the Windows NT/2000 Services from the desktop menu.

To list a service:

1. Go to the PUBLIC\ZENWORKS\WMINV\BIN directory.
2. At the prompt, enter **`ListSer [-verbose] "service_name"`**.

   where *service_name* refers to an Inventory service.

Follow the service naming syntax when you modify the service_name.

To refer to all services, use the asterisk (*) wildcard character within double quotes "*". This wildcard character can be used with ListSer parameters.

Inventory Upgrade Service

NOTE: Support Pack 1

The Inventory Upgrade Service runs the upgrade utilities for inventory. For example, this service makes the necessary updates in the database tables, and creates database tables. This service triggers full scan on workstations that have more than one MAC Addresses, IP/IPX Address, or DNS Name.

This service (Upgrade Service) is auto loaded on NetWare and Windows NT/2000 servers.

Understanding the Effects of Workstation Inventory Installation

On the servers, the Workstation Inventory installation program does the following:

- On NetWare servers:
  - Copies the inventory related files to the PUBLIC\ZENWORKS directory.
  - Copies the Workstation Inventory snap-in component to the PUBLIC\MGMT\CONSOLEONE*ConsoleOne_version*\BIN directory.
  - Assigns Create rights to the SCANDIR directory on NetWare servers.
  - The installation program assigns the SCANDIR as a trustee of the [Root] with Create rights.
  - Creates an Inventory Service object (*servername*_ZenInvservice) in NDS for each server on which Inventory is installed. This object is populated with the attributes: zeninvRole (role of the server), zeninvScanFilePath (path to SCANDIR directory), and zeninvHostServer (DN of the server on which Inventory is installed).
  - If the Inventory Service object already exists, the object is destroyed and created again.
  - During installation, the Inventory Service object is made a trustee of the NCP server with compare and read rights.
  - The installation program assigns the Inventory Service object as trustee to the tree.

    If there are any higher-level containers preceding the container that has the Inventory Service object, the Inventory Service object is assigned as a trustee of these containers.

    For example, if the Inventory tree is Novell_US, that has an Inventory Service object with the DN as us.california.sanjose.ZENInvService_SanJoseServer, where ZENInvService_SanJoseServer is the object, then ZENInvService_SanJoseServer is made a trustee of Novell_US, us container, california container, and sanjose container.

- Creates the scan directory (SCANDIR) with the subdirectories (ENTPUSHDIR, ENTMERGE, and DBDIR) in the specified volume on the server.
- The PUBLIC\ZENWORKS directory is made a trustee of [Root] with Read and filescan rights.

- On Windows NT/2000 servers:
  - Modifies the registry entries so that at run time, the scanners can be executed on the workstations from the servers. The installation adds the SYS share to the Registry entry on Windows NT/2000 servers.
  - Creates the SCANDIR directory with the subdirectories.
  - Creates an Inventory Service object (*servername*_ZenInvservice) in NDS for each server on which Inventory is installed. These attributes are populated: zeninvRole (Role of the server), zeninvScanFilePath (Path to ScanDir), and zeninvHostServer (DN of the server on which Inventory is installed).
  - If an Inventory Service object already exists, the object is destroyed and created again.
  - The installation program assigns the Inventory Service object as trustee to the tree.

    If there are any higher-level containers proceeding the container that has the Inventory Service object, the Inventory Service object is assigned as a trustee of these containers.

    For example, if the Inventory tree is Novell_US, that has an Inventory Service object with the DN as us.california.sanjose.ZENInvService_SanJoseServer, where ZENInvService_SanJoseServer is the object, then ZENInvService_SanJoseServer is made a trustee of Novell_US, us container, california container, and sanjose container.
  - Creates the SCANDIR directory (Share directory on the server\ZENWORKS\SCANDIR directory) with the subdirectories (ENTPUSHDIR, ENTMERGE, and DBDIR).
  - The installation program assigns rights to the SCANDIR directory. The installation program modifies the registry entries so that the Scanners can create the scan data files (.STR) in the SCANDIR directory.
  - On the server, the Inventory Service Manager is created as a service.

- On the database servers:
  - Installs the Sybase* ASA 7.0.0.313 database on the server you specify.
  - On a NetWare server, if Sybase 7 is already installed, only the database files are copied to the server.
  - If the Sybase database does not exist on the NetWare server, then Sybase database and database files are copied. Also, the installation program creates the MGMTDBS.NCF file in the SYS:\SYSTEM directory.
  - On Windows NT/2000 servers, Sybase database and the database files are copied. Also, the database server service (Adaptive Service Anywhere - ZENworks for Desktops 3) is created.
  - On the servers, the installation program initializes the database. At server startup time, the database is loaded.
  - If the MGMTDBS.NCF file already exists, then MGMTDB.DB and NAL.DB entries are added to this file, if these entries do not exist. Also, any other entries that ZfD3 uses are added if these entries do not exist.
  - Creates a Database object (ZfDInventoryDatabase) for Sybase and configures the properties of the object.
- On an existing ZENworks 2 installation:
  - On a NetWare server, detects whether a ZENworks 2 installation exists. If so, the existing directory under \SYS:\PUBLIC is renamed as ZENWORKS.ZEN2BKUP*currentdate*. Also, ZfD3 installation takes care of loading the Inventory database (Sybase) for ZENworks 2 and ZfD3.
  - Detects if an existing Sybase database with ZENworks 2 or ZENworks for Servers (ZfS) exists. If so, the installation program checks the database version and file size. If the existing Sybase version is 7.0.0.313, the installation program copies the new database and does not copy Sybase files.
  - If ZENworks 2 with Support Pack 1 exists on the server, the MGMTDB.DB and NAL.DB entries are added to MGMTDBS.NCF file. The ZENworks 2 entries in the AUTOEXEC.NCF entries will be deleted.

- If a ZENworks 2 setup exists on the server, the ZENINVDB entry is added to the MGMTDBS.NCF file. The ZENworks 2 entries in the AUTOEXEC.NCF entries will be deleted.
- ZENworks 2 inventory entries are removed from the AUTOEXEC.NCF file.

- On an existing ZfD3 installation:
  - Detects if ZfD3 files exist. If so, the files are overwritten. The existing Inventory Service object and the database object are overwritten. You need to configure the policies and properties for the new Inventory Service object and the database object.
- On the workstations:

  When you install the Novell client on the workstation, the following files are copied:

Filename	Directory
PUBLIC\ZENWORKS\WM95INV3.DLL	WINDOWS\SYSTEM on Windows 95/98
PUBLIC\ZENWORKS\WMINV3.DLL	WINNT\SYSTEM32 on Windows NT/2000
PUBLIC\ZENWORKS\INVSTAT.DLL	WINDOWS\SYSTEM on Windows 95/98
	WINNT\SYSTEM32 on Windows NT/2000
PUBLIC\ZENWORKS\LOC32VC.DLL	WINDOWS\SYSTEM on Windows 95/98
	WINNT\SYSTEM32 on Windows NT/2000
PUBLIC\ZENWORKS\NLS\ENGLISH\INVMSG.DLL	WINDOWS\SYSTEM\NLS\ENGLISH on Windows 95/98
	WINNT\SYSTEM32\NLS\ENGLISH on Windows NT/2000
PUBLIC\ZENWORKS\ZENPOL32.DLL	WINDOWS\SYSTEM on Win 95/98
	WINNT\SYSTEM32 on Win NT/2000

List of Inventory Components on Servers

Depending on the type of server, the following inventory components exist on the server.

Server Component	Root Server	Leaf Server with Database	Leaf Server	Intermediate Server	Intermediate Server with Database and Workstations	Intermediate Server with Database	Standalone Server	Root Server with Workstations
Service Manager	✓	✓	✓	✓	✓	✓	✓	✓
Selector	--	✓	✓	--	✓	--	✓	✓
Storer	✓	✓	--	--	✓	✓	✓	✓
Sender	--	✓	✓	✓	✓	✓	--	--
Receiver	✓	--	--	✓	✓	✓	--	✓
Database	✓	✓	--	--	✓	✓	✓	✓

Understanding ZENworks for Desktops 3 Inventory Attributes

NOTE: Support Pack 1

The following table lists the Workstation Inventory attributes that ZENworks for Desktops 3 uses.

Each row in the table has:

- Name of the attribute as displayed in the Inventory Database Export Wizard in ConsoleOne
- Name of the attribute in the exported CSV file (first row in the CSV)
- Inventory database attribute name
- Type of the attribute in the Inventory database
- Length of the attribute in the Inventory database
- Brief description of the attribute

For more information about the ZfD 3 Inventory database, see “Understanding the ZENworks for Desktops 3 Inventory Database Schema” on page 415.

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
General-NDSName-Label	NDSName_LABEL	ManageWise.NDSName.Label	String	254	The DN name of the workstation registered in NDS
General-Asset-Description	Asset_Description	ZENworks.SystemInfo.Description	String	254	Description of the system asset information
General-Asset-Caption	Asset_Caption	ZENworks.SystemInfo.Caption	String	64	Identifying information of the computer
General-Asset-Asset Tag	Asset_AssetTag	Zenworks.SytemInfo.AssetTag	String	256	Asset tag number that the ROM-based setup program creates. This is unique to every workstation
General-Asset-Model Number	Asset_ModelNumber	Zenworks.SystemInfo.Model	String	64	Model number value for the computer, assigned during manufacture
General-Asset-Serial Number	Asset_SerialNumber	Zenworks.SystemInfo.SerialNumber	String	64	Model serial number value for the computer, assigned during manufacture
General-User Information-Current Login User	UserInformation_Currentloginuser	ManageWise."User".CurrentLoginName	String	254	User logged in to the Primary NDS tree when the workstation was scanned
General-User Information.-Last Login User	UserInformation_Lastloginuser	ManageWise."User".LastLoginName	String	254	User logged in last to the Primary NDS tree when the workstation was scanned
SOFTWARE - Applications-Name	Applications_Name	CIM.Product.Name	String	256	Name of the software application

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SOFTWARE - Applications-Vendor	Applications_Vendor	CIM.Product. Vendor	String	256	Name of the software application manufacturer
SOFTWARE - Applications-Version	Applications_Version	CIM.Product. Version	String	64	Version of the software application
SOFTWARE-Operating Systems-Name	OperatingSystems_Name	ZENworks.ZENOperating System. OSType	Unsigned Small Integer (enum)		Operating system name. For example, Windows NT/Windows 2000 Enumeration values: 16 = "Microsoft Windows 95" 17 = "Microsoft Windows 98" 18 = "Microsoft Windows NT" 58 = "Microsoft Windows 2000"
SOFTWARE-Operating Systems-Version	OperatingSystems_Version	ZENworks.ZENOperating System. Version	String	254	Version of the operating system
SOFTWARE-Operating Systems-Description	OperatingSystems_Description	ZENworks.ZENOperating System. Description	String	254	More description about the operating system
SOFTWARE-Operating Systems-Install Date	OperatingSystems_InstallDate	ZENworks.ZENOperating System. InstallDate	String	25	Install date of the operating system

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SOFTWARE-Operating Systems-Size Stored in Paging Files	OperatingSystems_SizeStoredInPagingFiles(Kb)	ZENworks.ZENOperatingSystem. SizeStoredInPagingFiles	Unsigned Bit Integer		Total size allocated for paging by the operating system
SOFTWARE-Operating Systems-Code Page	OperatingSystems_CodePage	ZENworks.ZENOperatingSystem. CodePage	String	254	Current language code page being used
SOFTWARE-Scanner Information-Version	ScannerInformation_Version	ZENworks.InventoryScanner.Version	String	64	Version of the scanner running on the workstation
SOFTWARE-Scanner Information-Last Scan Date	ScannerInformation_LastScanDate	ZENworks.InventoryScanner. LastScanDate	Unsigned Bit Integer		The date when the scanner was last scanned. This is stored as milliseconds time value so that it could be read and displayed in any an appropriate date format
SOFTWARE-Scanner Information-Inventory Server	ScannerInformation_InventoryServer	ZENworks.InventoryScanner. InventoryServer	String	254	Name of the inventory server to which the scans are sent. It is not the complete DN of the server name
SOFTWARE-NetWare Client-Version	NetwareClient_Version	ZENworks.NetWareClient.Version	String	64	Version of the NetWare client software installed on the workstation

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SOFTWARE-Drivers-Network Adapter Driver-Description	NetworkAdapterDriver_Description	ZENworks.NetworkAdapterDriver.Description	String	254	Description of the Network adapter driver installed on the workstation. For example, IBM 10/100 Ethernet adapter, EN-2420Px Ethernet adapter
SOFTWARE-Drivers-Network Adapter Driver-Name	NetworkAdapterDriver_Name	ZENworks.NetworkAdapterDriver.Name	String	256	Name of the network adapter driver software installed, that corresponds to the adapter. For example, ne2000.sys, pppmac.vxd and others
SOFTWARE-Drivers-Network Adapter Driver-Version	NetworkAdapterDriver_Version	ZENworks.NetworkAdapterDriver. Version	String	64	Network adapter driver version
SOFTWARE-Drivers-Mouse Driver-Name	MouseDriver_Name	ZENworks.PointingDeviceDeviceDriver.Name	String	256	Name of the mouse driver installed on the workstation
SOFTWARE-Drivers-Mouse Driver-Version	MouseDriver_Version	ZENworks.PointingDeviceDeviceDriver.Version	String	64	Mouse driver version
HARDWARE-Mouse-Name	Mouse_Name	CIM.PointingDevice.Name	String	256	The name of the pointing device used here like *mouse*. In this case, it is Mouse. The string stored in this field will be *MOUSE*

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Mouse-Number of buttons	Mouse_NumberofButtons	CIM.PointingDevice.NumberOfButtons	Unsigned Tiny Integer		The number of buttons used by the pointing device
HARDWARE-Mouse-IRQ Number	Mouse_IRQNumber	CIM.IRQ.IRQNumber	Unsigned Integer		The IRQ channel on the system to which the *Mouse* pointing device is attached. This information is stored in an IRQ class and not in PointingDevice class in the database. For more information on how they are associated, see "Understanding the ZENworks for Desktops 3 Inventory Database Schema" on page 415
HARDWARE-Keyboard.Number of function keys	Keyboard_NumberofFunctionKeys	ZENworks.ZENKeyboard.NumberOfFunctionKeys	Unsigned Small Integer		Number of function keys on keyboard
HARDWARE-Keyboard.Layout	Keyboard_Layout	ZENworks.ZENKeyboard.layout	String	254	Layout information. For example, US English
HARDWARE-Keyboard.SubType	Keyboard_Subtype	ZENworks.ZENKeyboard.SubType	Unsigned Integer		A number indicating the subtype of the keyboard
HARDWARE-Keyboard.Delay	Keyboard_Delay(mSecs)	ZENworks.ZENKeyboard.Delay	Unsigned Integer		Delay before the repeat of a key
HARDWARE-Keyboard.Typematic rate	Keyboard_TypematicRate(mSecs)	ZENworks.ZENKeyboard.TypematicRate	Unsigned Integer		Rate of processing the keys

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Keyboard.Description	Keyboard_Description	ZENworks.ZENKeyboard.Description	String	254	Keyboard description indicating the type of keyboard. For example, IBM enhanced (101/102 key) keyboard
HARDWARE-Video BIOS.Manufacturer	VideoBIOS_Manufacturer	CIM.VideoBIOSElement.Manufacturer	String	254	Manufacturer of the video BIOS driver installed on the system
HARDWARE-Video BIOS.Version	VideoBIOS_Version	CIM.VideoBIOSElement.Version	String	254	Version of the Video BIOS driver
HARDWARE-Video BIOS.Install Date	VideoBIOS_InstallDate	CIM.VideoBIOSElement.InstallDate	String	25	Video BIOS release date
HARDWARE-Video BIOS. IS Shadowed	VideoBIOS_IsShadowed	CIM.VideoBIOSElement.ISShadowed	BIT (Used here for Boolean conditions)		Is a Boolean condition indicating the video BIOS supports Shadow memory. 0 represents false and 1 is true.
HARDWARE-Display Adapter.Number Of color panes	DisplayAdapter_NumberofColorPlanes	ZENworks.VideoAdpater.NumberOfColorPanes	Unsigned Integer		Number of color planes supported by the video system
HARDWARE-Display Adapter.Current Vertical Resolution	DisplayAdapter_CurrentVerticalResolution	ZENworks.VideoAdpater.CurrentVerticalResolution	Unsigned Integer		Vertical resolution of the display
HARDWARE-Display Adapter.Current Horizontal Resolution	DisplayAdapter_CurrentHorizontalResolution	ZENworks.VideoAdpater.CurrentHorizontalResolution	Unsigned Integer		Horizontal resolution of the display

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Display Adapter.Description	DisplayAdapter_Description	ZENworks.VideoAdpater.Description	String	254	Video adapter description
HARDWARE-Display Adapter.Min Refresh Rate	DisplayAdapter_Min_RefreshRate	ZENworks.VideoAdpater.MinRefreshRate	Unsigned Integer		Minimum refresh rate of the monitor for redrawing the display, measured in Hertz
HARDWARE-Display Adapter.Max Refresh Rate	DisplayAdapter_Max_RefreshRate	ZENworks.VideoAdpater.MaxRefreshRate	Unsigned Integer		Maximum refresh rate of the monitor for redrawing the display, measured in Hertz

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Display Adapter.Video Architecture	DisplayAdapter_VideoArchitecture	ZENworks.VideoAdpater.VideoArchitecture	Unsigned Integer (enum)		The architecture of the video subsystem in this system. For example, CGA/VGA/SVGA/8514A Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "CGA" 4 = "EGA" 5 = "VGA" 6 = "SVGA" 7 = "MDA" 8 = "HGC" 9 = "MCGA" 10 = "8514A" 11 = "XGA" 12 = "Linear Frame Buffer" 160 = "PC-98"
HARDWARE-Display Adapter.Video Memory Type	DisplayAdapter_VideoMemoryType	ZENworks.VideoAdpater.VideoMemoryType	Unsigned Small Integer		The type of memory for this adapter. For example, VRAM/SRAM/DRAM/EDO RAM
HARDWARE-Display Adapter.Max memory supported	DisplayAdapter_MaxMemorySupported(bytes)	ZENworks.VideoAdpater.MaxMemorySupported	Unsigned Integer		Maximum memory that the display adapter supports for VIDEO RAM

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Display Adapter.Current Bits/Pixel	DisplayAdapter_CurrentBits/Pixel	ZENworks.VideoAdpater.CurrentBitsPerPixel	Unsigned Integer		Number of adjacent color bits for each pixel
HARDWARE-Modem.Caption	Modem_Caption	CIM.POTSModem.Caption	String	64	Modem label. Usually the name of the manufacturer is stored here. For example, 3Com, IBM
HARDWARE-Modem.Description	Modem_Description	CIM.POTSModem.Description	String	254	The complete description of the modem. For example, Standard 2400 bps modem, IBM PCMCIA HPC modem
HARDWARE-Modem.Name	Modem_Name	CIM.POTSModem.Name	String	256	The name of the modem dictating its type and usage. For example, Standard Windows Modem means that this is used in standard windows architecture
HARDWARE-BIOS.BIOS Identification Bytes	BIOS_BIOSIdentificationBytes	ZENworks.BIOS.BIOSIDBytes	String	254	Byte in the BIOS that indicates the computer model
HARDWARE-BIOS.Serial Number	BIOS_SerialNumber	ZENworks.BIOS.SerialNumber	String	64	Serial number of BIOS assigned by the manufacturer
HARDWARE-BIOS.Primary BIOS	BIOS_PrimaryBios	ZENworks.BIOS.PrimaryBIOS	BIT (Used for Boolean conditions here)		True when set to 1, indicates that this BIOS is the primary BIOS. Used in systems with additional BIOS chips

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-BIOS.Install Date	BIOS_InstallDate	ZENworks.BIOS.InstallDate	String	25	The release date of the BIOS given by the manufacturer
HARDWARE-BIOS.Version	BIOS_Version	ZENworks.BIOS.Version	String	254	Version or revision level of the BIOS
HARDWARE-BIOS.Manufacturer	BIOS_Manufacturer	ZENworks.BIOS.Manufacturer	String	254	The manufacturer name of BIOS
HARDWARE-BIOS.Caption	BIOS_Caption	ZENworks.BIOS.Caption	String	64	The name of the BIOS as given by the BIOS manufacturer
HARDWARE-BIOS.size	BIOS_Size(KB)	ZENworks.BIOS.size	Unsigned Integer		Size of the BIOS in bytes
HARDWARE-Processor-Maximum clock speed	Processor_MaximumClockSpeed(MHz)	CIM.Processor.MaxClockSpeed	Unsigned Integer		Maximum clock speed of the processor in MHz
HARDWARE-Processor-Role	Processor_Role	CIM.Processor.Role	String	254	Type of processor such as central processor, math coprocessor, and others

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Processor-Processor Family	Processor_ProcessorFamily	CIM.Processor.Family	Unsigned Small Integer (enum)		Identification of the processor family such as Pentium* II, Pentium III, and others Enumeration Values: 1 = "Other" 2 = "Unknown" 3 = "8086" 4 = "80286" 5 = "80386" 6 = "80486" 7 = "8087" 8 = "80287" 9 = "80387" 10 = "80487" 11 = "Pentium Family" 12 = "Pentium Pro" 13 = "Pentium II" 14 = "Pentium MMX" 15 = "Celeron" 16 = "Xeon" 18 = "M1 Family" 19 = "M2 Family" 25 = "K5 Family" 26 = "K6 Family"

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
					32 = "Power PC Family" 33 = "Power PC 601" 34 = "Power PC 603" 35 = "Power PC 603+" 36 = "Power PC 604" 48 = "Alpha Family" 64 = "MIPS Family" 80 = "SPARC Family" 96 = "68040" 97 = "68xxx Family" 98 = "68000" 99 = "68010" 100 = "68020" 101 = "68030" 112 = "Hobbit Family" 128 = "Weitek" 144 = "PA-RISC Family" 160 = "V30 Family"
HARDWARE-Processor-Other family description	Processor_OtherFamilyDescription	CIM.Processor.OtherFamilyDescription	String	64	Additional description about the processor family, such as the Pentium processor with MMX technology
HARDWARE-Processor-Upgrade Method	Processor_UpgradeMethod	CIM.Processor.UpgradeMethod	Unsigned Small Integer		The method by which this processor can be upgraded, if upgrades are supported

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Processor - Processor Stepping	Processor_ProcessorStepping	CIM.Processor.Stepping	String	254	Single-byte code characteristic provided by microprocessor vendors to identify the processor stepping model
HARDWARE-Processor-Device ID	Processor_DeviceID	CIM.Processor.DeviceID	String	64	Special Hexadecimal string identifying the processor type
HARDWARE-Memory-Physical Memory-Physical Memory Size	PhysicalMemory_PhysicalMemorySize(Kb)	CIM.Memory.NumberOfBlocks	Unsigned Bit Integer		Total physical memory size of the workstation
HARDWARE-Memory-Virtual Memory-Total Virtual Memory	VirtualMemory_TotalVirtualMemory(Kb)	ZENworks.VirtualMemory.TotalVirtualMemorySize			Indicates the total number of bytes in the virtual address space
HARDWARE-Memory-Cache Memory-Speed	CacheMemory_Speed(nsec)	CIM.PhysicalMemory.Speed	Unsigned Bit Integer		Speed of this System Cache module in nanoseconds. This is stored in CIM.PhysicalMemory class and is associated to CIM.CacheMemory. For more information on how they are associated, see "Understanding the ZENworks for Desktops 3 Inventory Database Schema" on page 415

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Capacity	CacheMemory_Capacity(Kb)	CIM.PhysicalMemory.Capacity	Unsigned Bit Integer		Capacity of this System Cache module in nanoseconds. This is stored in CIM.PhysicalMemory class and is associated to CIM.CacheMemory. For more information on how they are associated, see “Understanding the ZENworks for Desktops 3 Inventory Database Schema” on page 415
HARDWARE-Memory-Cache Memory-Level	CacheMemory_Level	CIM.CacheMemory."Level"	Unsigned Small Integer (enum)		Indicates the cache level; internal cache that is built in to the microprocessors; external cache that is between the CPU and DRAM Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Primary" 4 = "Secondary" 5 = "Tertiary"

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Write Policy	CacheMemory_WritePolicy	CIM.CacheMemory.WritePolicy	Unsigned Small Integer (enum)		Indicates the two different ways (Write-Back and Write-Through Cache) that the cache can handle to write to the memory Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Write Back" 4 = "Write Through" 5 = "Varies with Address"
HARDWARE-Memory-Cache Memory-Error methodology	CacheMemory_ErrorMethodology	CIM.CacheMemory. ErrorMethodology	String	254	Error correction scheme supported by this cache component, for example, Parity/Single Bit ECC/ MultiBit ECC
HARDWARE-Memory-Cache Memory-Cache type	CacheMemory_CacheType	CIM.CacheType	Unsigned Small Integer (enum)		Defines the system cache type. For example, Instruction, Data, Unified. Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Instruction" 4 = "Data" 5 = "Unified"
HARDWARE-Memory-Cache Memory-Line Size	CacheMemory_LineSize(Bytes)	CIM.CacheMemory.LineSize	Unsigned Integer		Size in bytes of a single cache bucket or line

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Replacement Policy	CacheMemory_ReplacementPolicy	CIM.CacheMemory.ReplacementPolicy	Unsigned Integer (enum)		Algorithm that the cache uses to determine which cache lines or buckets should be reused. Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Least Recently Used (LRU)" 4 = "First In First Out (FIFO)" 5 = "Last In First Out (LIFO)" 6 = "Least Frequently Used (LFU)" 7 = "Most Frequently Used (MFU)"
HARDWARE-Memory-Cache Memory-Read Policy	CacheMemory_ReadPolicy	CIM.CacheMemory.ReadPolicy	Unsigned Small Integer (enum)		Indicates whether the data cache is for read operation. Enumeration Values: 1 = "Other" 2 = "Unknown" 3 = "Read" 4 = "Read-ahead" 5 = "Read and Read-ahead"

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Memory-Cache Memory-Associativity	CacheMemory_Associativity	CIM.CacheMemory.Associativity	Unsigned Integer (enum)		Defines the system cache associativity (direct-mapped, 2-way, 4-way) Enumeration values: 1 = "Other" 2 = "Unknown" 3 = "Direct Mapped" 4 = "2-way Set-Associative" 5 = "4-way Set-Associative" 6 = "Fully Associative"
HARDWARE-Disk-Floppy-Manufacturer	Floppy_Manufacturer	ZENworks.PhysicalDiskette.Manufacturer	String	256	Vendor name
HARDWARE-Disk- Floppy-.Description	Floppy_Description	ZENworks.PhysicalDiskette. Description	String	254	Floppy diskette description
HARDWARE-Disk-Floppy-Physical Cylinders	Floppy_PhysicalCylinders	ZENworks.PhysicalDiskette.PhysicalCylinders	Unsigned Integer		Total number of cylinders or tracks on the floppy
HARDWARE-Disk- Floppy - Physical Heads	Floppy_PhysicalHeads	ZENworks.PhysicalDiskette.PhysicalHeads	Unsigned Small Integer		Number of heads
HARDWARE-Disk- Floppy - Capacity	Floppy_Capacity(Kb)	ZENworks.PhysicalDiskette.Capacity	Unsigned Bit Integer		Total size

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Disk- Floppy - Sectors/Track	Floppy_Sectors/ Track	ZENworks.PhysicalDiskette.SectorsPerTrack	Unsigned Integer		Number of sectors per track
HARDWARE-Disk-Floppy Drive- DeviceID	FloppyDrive_DeviceID	CIM.Diskette Drive	String	64	The drive name representing the floppy drive
HARDWARE-Disk-Hard Disk-Manufacturer	HardDisk_Manufacturer	ZENworks.PhysicalDisk. Manufacturer	String	256	Vendor name
HARDWARE-Disk- Hard Disk - .Description	HardDisk_Description	ZENworks.PhysicalDisk. Description	String	254	Hard disk vendor description
HARDWARE-Disk- Hard Disk - Physical Cylinders	HardDisk_PhysicalCylinders	ZENworks.PhysicalDisk.PhysicalCylinders	Unsigned Integer		Total number of cylinders
HARDWARE-Disk- Hard Disk - Physical Heads	HardDisk_PhysicalHeads	ZENworks.PhysicalDisk.PhysicalHeads	Unsigned Small Integer		Number of heads
HARDWARE-Disk- Hard Disk - Sectors/Track	HardDisk_Sectors/Track	ZENworks.PhysicalDisk.SectorsPerTrack	Unsigned Integer		Number of sectors per track
HARDWARE-Disk- Hard Disk - Capacity	HardDisk_Capacity(Kb)	ZENworks.PhysicalDisk.Capacity	Unsigned Bit Integer		Total size of the hard disk
HARDWARE-Disk-Hard Disk Drive-Device ID	HardDiskDrive_DeviceID	ZENworks.LogicalDiskDrive. DeviceID	String	64	The logical drive letter partitioned on the hard disk. For example, C:, D:, and others

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Disk-Hard Disk Drive-Size	HardDiskDrive_Size(MB)	ZENworks.LogicalDiskDrive.TotalSize	Unsigned Integer		Total size on the logical drive
HARDWARE-Disk-Hard Disk Drive-Volume Serial Number	HardDiskDrive_VolumeSerialNumber	ZENworks.LogicalDiskDrive. VolumeSerialNumber	String	254	Volume serial number of the logical drive
HARDWARE-Disk-Hard Disk Drive-Caption	HardDiskDrive_Caption	ZENworks.LogicalDiskDrive.Caption	String	64	Volume label assigned to the logical drive
HARDWARE-Disk-CDROM-Name	CDROM_Name	ZENworks.PhysicalCDROM.Manufacturer	String	256	The manufacturer of the CD-ROM drive
HARDWARE-Disk-CDROM-Caption	CDROM_Caption	ZENworks.PhysicalCDROM.Caption	String	64	CDROM label
HARDWARE-Disk-CDROM-Description	CDROM_Description	ZENworks.PhysicalCDROM.Description	String	256	Description of the CDROM drive, as given by the manufacturer. For example, ATAPI CDROM, CREATIVE CD1620E SL970520
HARDWARE-Disk-CDROM Drive-DeviceID	CDROMDrive_DeviceID	ZENworks.LogicalCDROM.DeviceID	String	64	Drive letter allocated for the CD-ROM on the workstation
HARDWARE-Disk-Backup Disk-Name	BackUpDisk_Name	ZENworks.SCSIDrive.Name	String	256	Name of the backup device vendor. For example, IOMEGA

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Disk-Backup Disk-Description	BackUpDisk_Description	ZENworks.SCSIDrive.Description	String	254	The description of the backup device or type. For example, IOMEGA ZIP 100, indicates it is a zip drive
HARDWARE-Disk-Backup Disk Drive-DeviceID	BackUpDiskDrive_DeviceID	ZENworks.LogicalSCSIDrive-DeviceID	String	64	The drive mapped to the backup disk on the workstation
HARDWARE-Ports-Serial Port-Name	SerialPort_Name	ZENworks.SerialPort.Name	String	256	The name of the serial port. For example, COM1, COM2 and others
HARDWARE-Ports-Serial Port-Address	SerialPort_Address	ZENworks.SerialPort.Address	Unsigned Bit Integer		The address mapped in memory for the serial port
HARDWARE-Ports-Serial Port-IRQ	SerialPort_IRQ	CIM.IRQ.IRQ Number	Unsigned Integer		The IRQ channel on the system to which the serial port is attached. In the database, this information is stored in an IRQ class and not in Serial Port class. For more information on how they are associated, see "Understanding the ZENworks for Desktops 3 Inventory Database Schema" on page 415
HARDWARE-Ports-Parallel Port-Name	ParallelPort_Name	ZENworks.ParallelPort.Name	String	256	The name of the parallel port. For example, LPT1, LPT1 and others
HARDWARE-Ports-Parallel Port-Address	ParallelPort_Address	ZENworks.ParallelPort.Address	Unsigned Bit Integer		The name of the parallel port. For example, LPT1, LPT1 and others

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Ports-Parallel Port-DMA Support	ParallelPort_DMA Support	ZENworks.ParallelPort.DMASupport	BIT (used for Boolean conditions here)		If true or 1, then it means that DMA is channel is allocated for bulk data transfer for use with devices connected to the parallel ports
HARDWARE-Ports-Parallel Port-IRQ	ParallelPort_IRQ	CIM.IRQ.IRQ Number	Unsigned Integer		The IRQ channel on the system to which the parallel port is attached. This information is stored in an IRQ class and not in parallel Port class in the database. For more information on how they are associated, see "Understanding the ZENworks for Desktops 3 Inventory Database Schema" on page 415
HARDWARE-Bus-Version	Bus_Version	ZENworks.Bus.BusVersion	String	254	Version of the Bus supported by the motherboard
HARDWARE-Bus-Protocol Supported	Bus_ProtocolSupported	ZENworks.Bus.ProtocolSupported	Unsigned Small Integer		Bus type indicates PCI, ISA, and others. Enumeration Values: 2 = "UNKNOWN" 3 = "EISA" 4 = "ISA" 5 = "PCI"
HARDWARE-Network Adapter-Name	NetworkAdapter_Name	CIM.EthernetAdapter.Name	String	256	Network adapters installed on the system

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Network Adapter-Max. Speed	NetworkAdapter_Max_Speed(bits/sec_)	CIM.EthernetAdpater.MaxSpeed	Unsigned Bit Integer		Rate at which the adapter can transfer data
HARDWARE-Network Adapter-Permanent Address	NetworkAdapter_PermanentAddress	CIM.EthernetAdapter.PermanentAddress	String	64	Machine address stored permanently in the adapter (MAC address)
HARDWARE-MultiMedia Card-Description	MultimediaCard_Description	ZENworks.SoundCard.Description	String	254	Description of the multimedia component for the workstation
HARDWARE-MultiMedia Card-Name	MultimediaCard_Name	ZENworks.SoundCard.Name	String	256	Name of the sound card installed on the system
HARDWARE-MultiMedia Card-Manufacturer	MultimediaCard_Manufacturer	ZENworks.SoundCard.Manufacturer	String	256	Vendor name
HARDWARE-Cards-Description	Cards_Description	CIM.Card.Description	String	254	Name of the cards installed on the workstation
HARDWARE-Battery-Name	Battery_Name	CIM.Battery.Name	String	254	Name of the battery installed on the system

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Battery-Chemistry	Battery_Chemistry	CIM.Battery.Chemistry	Unsigned Small Integer		Indicates battery's chemistry such as Lead Acid, Nickel Cadmium and others. Enumeration Values: 1 = "Other" 2 = "Unknown" 3 = "Lead Acid" 4 = "Nickel Cadmium" 5 = "Nickel Metal Hydride" 6 = "Lithium-ion" 7 = "Zinc air" 8 = "Lithium Polymer"
HARDWARE-Battery-Design Capacity	Battery_DesignCapacity(mWatt-hours)	CIM.Battery.DesignCapacity	Unsigned Integer		The design capacity of the battery in mWatt-hours
HARDWARE-Battery-Design Voltage	Battery_DesignVoltage(MilliVolts)	CIM.Battery.DesignVoltage	Unsigned Bit Integer		The design voltage of the battery in mVolts
HARDWARE-Battery-Smart Battery Version	Battery_SmartBatteryVersion	CIM.Battery.SmartBatteryVersion	String	64	The Smart Battery Data Specification version number supported by this battery
HARDWARE-Battery-Manufacturer	Battery_Manufacturer	CIM.PhysicalComponent. Manufacturer	String	254	Vendor name of the battery
HARDWARE-Battery-Install Date	Battery_InstallDate	CIM.PhysicalComponent. InstallDate	String	25	Date of manufacturing the battery

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
HARDWARE-Battery-Serial Number	Battery_SerialNumber	CIM.PhysicalComponent.SerialNumber	String	64	Battery serial number
HARDWARE-Power Supply-Description	PowerSupply_Description	CIM.PowerSupply.Description	String	254	Name and description of the power supply on the system
HARDWARE-Power Supply-Total Output Power	PowerSupply_TotalOutputPower(MilliWatts)	CIM.PowerSupply.TotalOutputPower	Unsigned Integer		Total output power of the power supply
NETWORK-IP Address - Address	IPAddress_Address	CIM.IPProtocolEndpoint.Address	String	254	IP address of the workstation
NETWORK-IP Address – Subnet Mask	IPAddress_SubnetMask	CIM.IPProtocolEndpoint.SubnetMask	String	254	The subnet mask of the workstation
NETWORK – DNS - LABEL	DNS_LABEL	ManageWise.DNSName.Label	String	254	DNS name of the workstation
NETWORK -IPX Address - Address	IPXAddress_Address	CIM.IPXProtocolEndpoint.Address	String	254	IPX address of the workstation
NETWORK -Mac Address – MACAddress	MacAddress_MACAddress	CIM.LANEndpoint.MACAddress	String	12	MAC address of the workstation
SYSTEM-MotherBoard-Hosting Board	MotherBoard_HostingBoard	CIM.Card.HostingBoard	BIT (used here for Boolean conditions)		When true or 1, the card represents a mother board. For all other cards, this attributes will be false or 0

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SYSTEM-MotherBoard-Version	MotherBoard_Version	CIM.Card.Version	String	64	Motherboard version
SYSTEM-MotherBoard-Description	MotherBoard_Description	CIM.Card.Description	String	254	The number of expansion slots on the motherboard
SYSTEM-IRQ-IRQ Number	IRQ_IRQNumber	CIM.IRQ.IRQNumber	Unsigned Integer		The system interrupt number
SYSTEM-IRQ-Availability	IRQ_IRQAvailability	CIM.IRQ.Availability	Unsigned Small Integer		Indicates whether the IRQ channel is used or available
SYSTEM-IRQ-IRQ Trigger Type	IRQ_IRQTriggerType	CIM.IRQ.TriggerType	Unsigned Small Integer		IRQ trigger type indicating whether edge (value=4) or level triggered (value=3) interrupts occur
SYSTEM-IRQ-IRQ Shareable	IRQ_IRQShareable	CIM.IRQ.Shareable	Unsigned Small Integer		Boolean indicating whether the IRQ can be shared
SYSTEM-SLOT-Max. Data Width	Slot_Max_DataWidth	CIM.Slot.MaxDataWidth	Unsigned Small Integer		Maximum bus width of adapter cards that can be inserted into this slot in bits. If the value is 'unknown', enter 0. If the value is other than 8, 16, 32, 64 or 128, enter 1. It is expressed in bits
SYSTEM-SLOT-Thermal Rating	Slot_ThermalRating(MilliWatts)	CIM.Slot.ThermalRating	Unsigned Integer		Maximum thermal dissipation of the slot in milliwatts
SYSTEM-DMA-DMA Channel Number	DMA_DMAChannelNumber	CIM.DMA.DMAChannel	Unsigned Integer		The DMA channel number

Export Wizard Attribute Name	Export Attribute Name (Column Heading in the CSV file)	Database Schema Attribute Name	Data Type	Length	Description of the Attribute
SYSTEM-DMA-Description	DMA_Description	CIM.DMA.Description	String	254	The name of the device using the DMA channel
SYSTEM-DMA-Availability	DMA_IRQAvailability	CIM.DMA.Availability	Unsigned Small Integer		Indicates whether the DMA channel is available or not
SYSTEM-DMA-DMA Burst Mode	DMA_DMABurstMode	CIM.DMA.BurstMode	BIT (used here for Boolean condition)		Indication that the DMA channel supports the burst mode

Understanding the ZENworks for Desktops 3 Inventory Database Schema

This section describes the design of the ZfD3 Inventory database schema implemented using the Common Information Model (CIM) of Distributed Management Task Force (DMTF). To understand this document effectively, you should be familiar with terminology such as CIM and Desktop Management Interface (DMI). You should also have a solid understanding of Relation Database Based Managed Systems (RDBMS) and database concepts.

The following sections provide in-depth information:

- "Overview" on page 415
- "CIM Schema" on page 416
- "Inventory Database Schema in ZfD3" on page 430

Overview

The DMTF is the industry organization leading the development, adoption, and unification of management standards and initiatives for desktop, enterprise, and Internet environments. See the DMTF Web site (http://www.dmtf.org) for more information about DMTF.

The DMTF CIM is an approach to system and network management that applies the basic structuring and conceptualization techniques of the object-oriented paradigm. The approach uses a uniform modeling formalism that together with the basic repertoire of object-oriented constructs supports the cooperative development of an object-oriented schema across multiple organizations.

A management schema is provided to establish a common conceptual framework at the level of a fundamental topology, both with respect to classification and association, and to a basic set of classes intended to establish a common framework for a description of the managed environment. The management schema is divided into the following conceptual layers:

- **Core Model:** An information model that captures notions that are applicable to all areas of management.
- **Common Model:** An information model that captures notions that are common to particular management areas, but independent of a particular technology or implementation. The common areas are systems, applications, databases, networks, and devices. The information model is specific enough to provide a basis for the development of management applications. This model provides a set of base classes for extension into the area of technology-specific schemas. The Core and Common models together are expressed as the CIM schema.
- **Extension Schemas:** This schema represents technology-specific extensions of the Common model. These schemas are specific to environments, such as operating systems, for example, UNIX* or Microsoft Windows.

CIM comprises a specification and a schema (see the DMTF Web site (http://www.dmtf.org/spec/cims.html). The specification defines the meta-schema plus a concrete representation language called Managed Object Format (MOF).

CIM Schema

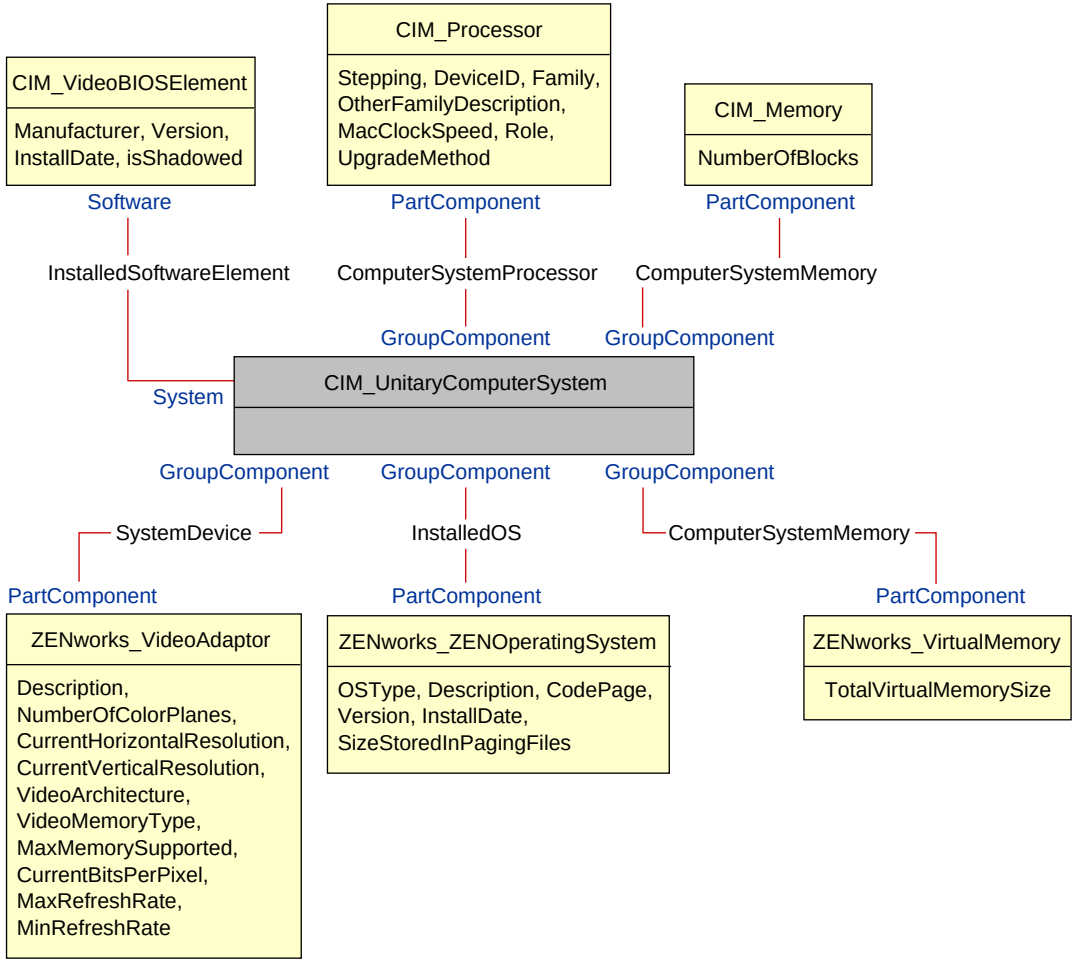
The elements of the meta schema are classes, properties, and methods. The meta schema also supports indications and associations as types of classes and references as types of properties.

Classes can be arranged in a generalization hierarchy that represents subtype relationships between classes. The generalization hierarchy is a rooted, directed graph that does not support multiple inheritance.

A regular class may contain scalar or array properties of any intrinsic type such as Boolean, integer, string, and others. It may not contain embedded classes or references to other classes.

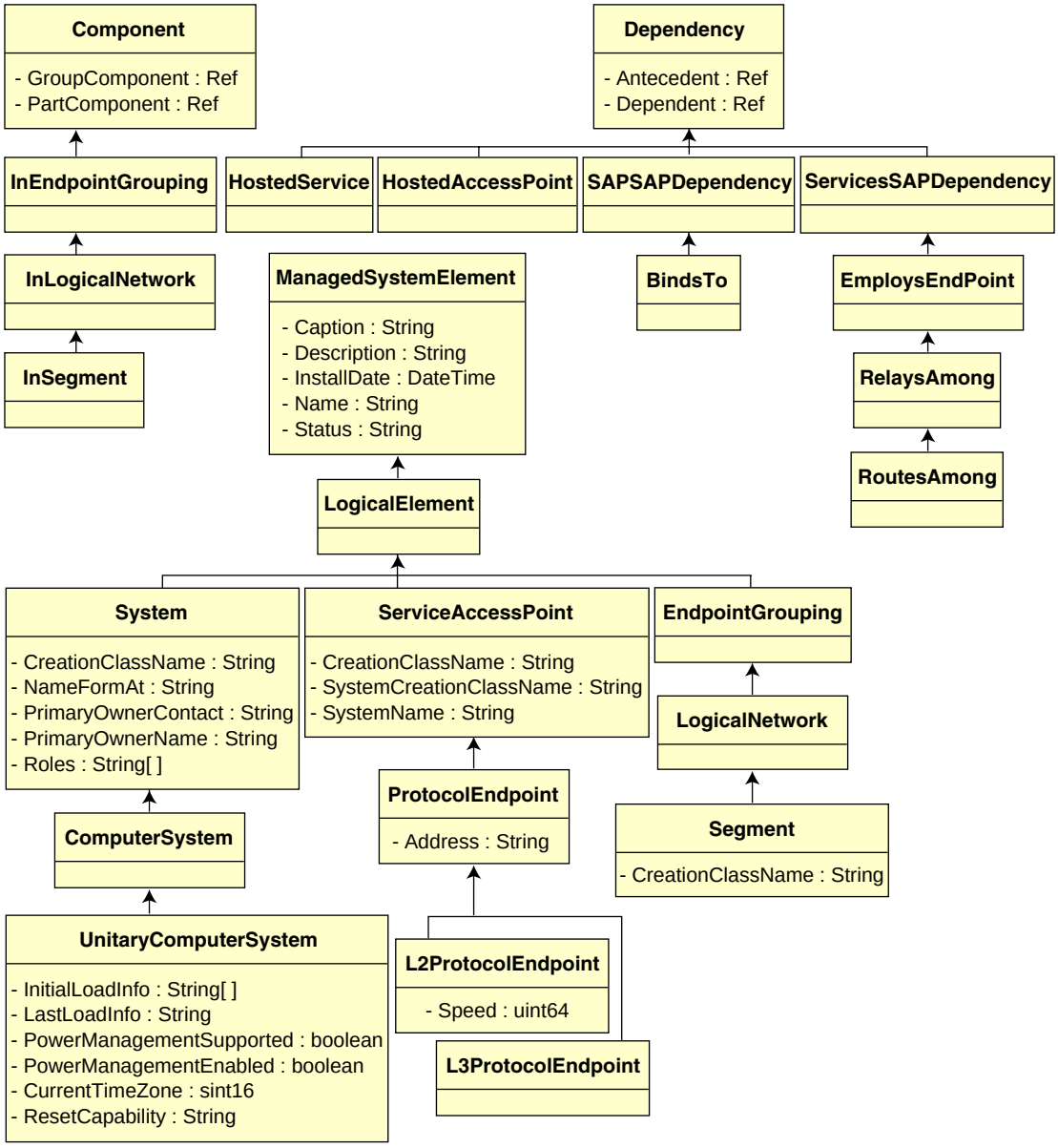
An association is a special class that contains two or more references. It represents a relationship between two or more objects. Because of the way associations are defined, it is possible to establish a relationship between classes without affecting any of the related classes. That is, addition of an association does not affect the interface of the related classes. Only associations can have references.

The schema fragment in the following illustration shows the relationships between some CIM objects that ZfD uses:



The illustration shows how the CIM schema maps to a relational DBMS schema. The classes are shown with the class name as the box heading. The associations are labeled within the lines between two classes.

The inheritance hierarchy of this schema fragment is shown in the following illustration of the CIM 2.2 schema. The references shown as type Ref are in bold with each association sub-type narrowing the type of the reference.



CIM-to-Relational Mapping

CIM is an object model complete with classes, inheritance, and polymorphism. The generated mapping to a relational schema preserves these

features to the maximum extent. The following two aspects are part of the relational mapping:

- **Logical Schema:** The logical schema defines how the data appears to applications, similar to an API. The goal is that the logical schema remains the same irrespective of the underlying database so that application software can run unchanged on any supported databases. Though SQL is a standard, this goal is not fully possible. Application software will need to know more about the database in use and this information can be abstracted and isolated to a small area of the application code.
- **Physical Schema:** The physical schema defines how the data is structured in the database. The schema tends to be specific to the database because of the nature of SQL and RDBMs. This document will describe the physical schema in general terms only.

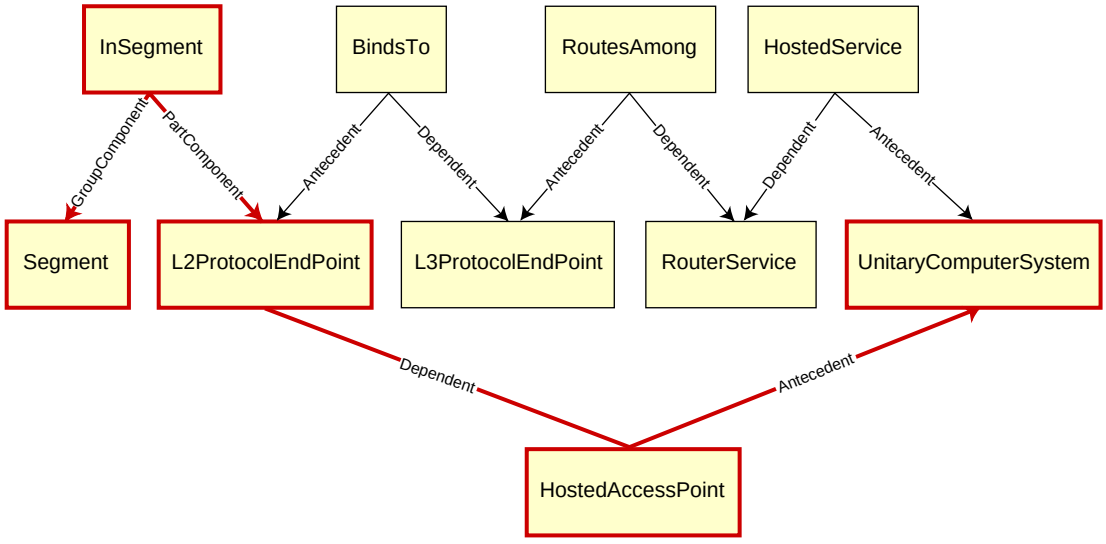
A table in the database represents each class in the CIM hierarchy. A column of the appropriate type in the table represents each non-inherited property in the class. Each table also has a primary key, id$, which is a 64-bit integer that uniquely identifies an instance. An instance of a CIM class is represented by a row in each table that corresponds to a class in its inheritance hierarchy. Each row has the same value for id$.

Each CIM class is also represented by a view that uses id$ to join rows from the various tables in the inheritance hierarchy to yield a composite set of properties (inherited plus local) for an instance of that class. The view also contains an extra column, class$, of type integer that represents the type of the actual (leaf-most) class of the instance.

Associations are mapped in the same manner as regular classes, with a reference property being represented by a column with the id$ field of the referenced object instance. Thus associations can be traversed by doing a join between the reference field in the association and the id$ field in the referenced table. The following illustration depicts a typical query using this mapping:

Get Computers for Segment

```
SELECT CIM.UnitaryComputerSystem.*
FROM    CIM.UnitaryComputerSystem, CIM.Segment, CIM.L2ProtocolEndPoint,
        CIM.HostedAccessPoint, CIM.InSegment
WHERE  CIM.SegmentName = 'xxx'
AND     CIM.InSegment.GroupComponent = CIM.Segment.id$
AND     CIM.InSegment.PartComponent = CIM.L2ProtocolEndPoint.id$
AND     CIM.HostedAccessPoint.Dependent = CIM.L2ProtocolEndPoint.id$
AND     CIM.HostedAccessPoint.Antecedent = CIM.UnitaryComputerSystem.id$
```



This query finds all the computers attached to a given network segment. The classes and relationships involved are highlighted with borders.

The following topics describe both the schema types:

- “Logical Schema” on page 421
- “Physical Schema” on page 429

Logical Schema

The logical schema is the database schema as seen by users of the database and the application program. The schema consists of stored procedures and views. The underlying tables are not visible to the application.

Typically, each CIM class has the following:

- A constructor procedure to generate an instance of the class. See “Constructor” on page 426 for more information.
- A destructor procedure to destroy an instance of the class. See “Destructor” on page 429 for more information.
- A view to access and update the values of properties of the class.

ZfD Inventory components use JDBC to issue SQL statements to the RDBMS and to convert between RDBMS data types and Java data types. The use of JDBC with stored procedures and views provides a level of abstraction that insulates application code from the underlying database technology and from changes to the physical schema.

The various elements of the logical schema are discussed in more detail in the following sections:

- “Naming Schema Elements” on page 422
- “Users and Roles” on page 423
- “Data Types” on page 424
- “Views” on page 424
- “Object Identifier Id$” on page 425
- “Constructor” on page 426
- “Destructor” on page 429

Naming Schema Elements

We recommend that you use the CIM names unchanged in the database schema. Some problems may still ensue because of the differences in the naming schemes, such as the following:

- Names in CIM and SQL are not case-sensitive.
- All databases have different sets of reserved words that must be enclosed in quotes (" ") when used as schema element names; however, in Oracle, enclosing a name in quotes makes it case-sensitive.
- CIM classes avoid using SQL reserved words as names.
- CIM names are not limited in length and usually the names are long. Sybase allows up to 128 characters, but Oracle* restricts the names to 30 characters.

Most of these problems are avoided during schema generation by preserving the case of CIM names, abbreviating any names longer than 30 characters, and placing quotes around any name that is in the union of the sets of reserved words.

Any name longer than 28 characters is abbreviated to a root name of 28 or fewer characters to allow a two-character prefix so that all associated SQL schema elements can use the same root name. The abbreviation algorithm shortens a name so that it is mnemonic, recognizable, and also unique within its scope. The abbreviated name is given a # character as a suffix (note that # is an illegal character in CIM) to prevent clashes with other names. If two or more names within the same scope generate the same abbreviation, an additional digit is appended to make the name unique. For example, AttributeCachingForRegularFilesMin is abbreviated to AttCacForRegularFilesMin#.

All such mangled names are written to the mangled name table so that a program can look up the real CIM name and retrieve the mangled name to use with the SQL.

Views are the schema elements that are most often manipulated by application code and queries. They use the same name as the CIM class they represent. For example, the CIM_UnitaryComputerSystem class is represented by a view named CIM.UnitaryComputerSystem.

When necessary, names for indexes and auxiliary tables are created by concatenating the class name and property name separated by a $ character. These names are usually abbreviated. For example, NetworkAdapter$NetworkAddresses is abbreviated to NetAdapter$NetAddresses#. This does not have any adverse impact on ZfD3 schema users.

Users and Roles

In SQL, a user with the same name as the schema is the owner of each schema, for example, CIM, ManageWise®, ZENworks, and others.

Additionally, there is an MW_DBA user that has Database Administrator privileges and rights to all schema objects. The MW_Reader role has read-only access to all schema objects and the MW_Updater role has read-write-execute access to all schema objects.

Application programs should access the database as either MW_Reader or MW_Updater for a Sybase database and MWO_Reader or MWO_Updater for an Oracle database, depending on their requirements.

Data Types

CIM data types are mapped to the most appropriate data type provided by the database. Usually, the Java application does not require the type because it uses JDBC to access the data.

Java does not natively support unsigned types, so you should use classes or integer types of the next size to represent them. Also, ensure that there are no problems while reading or writing to the database. For example, reading or writing a negative number to an unsigned field in the database is likely cause an error.

Strings in CIM and Java are Unicode*, so the database is created using the UTF8 character set. Internationalization does not pose any problem; however, it may create problem with case sensitivity in queries.

All databases preserve the case of string data stored within them, but may access the data as either case sensitive or otherwise during queries. In ZfD, the Inventory Query component is not affected because the queried data is retrieved from the database before being queried and so case sensitivity is automatically taken care of.

In CIM, strings may be specified with or without a maximum size in characters. Many strings have no specified size, which means they can be unlimited in size. For efficiency reasons, these unlimited strings are mapped to a variable string with maximum size of 254 characters. CIM strings with a maximum size are mapped to variable database strings of the same size. The size in the database is in bytes and not as characters because a Unicode character may require more than one byte for storage.

Views

Each CIM class is represented in the database by a view that contains all the local and inherited non-array properties of that class. The view is named the same as the CIM class. For example, the CIM class CIM_System represents a SQL view named CIM.System, as shown in the following illustration:

```
CREATE VIEW CIM.System
(
   id$,
   class$,
   Caption,
   Description,
   InstallDate,
   Status,
   CreationClassName,
   Name,
   NameFormat,
   PrimaryOwnerContact,
   PrimaryOwnerName
)
AS SELECT
   CIM.t$ManagedSystemElement.id$,
   MW_DBA.extractClass(CIM.t$ManagedSystemElement.id$),
   CIM.t$ManagedSystemElement.Caption,
   CIM.t$ManagedSystemElement.Description,
   CIM.t$ManagedSystemElement.InstallDate,
   CIM.t$ManagedSystemElement.Status,
   CIM.t$System.CreationClassName,
   CIM.t$ManagedSystemElement.Name,
   CIM.t$System.NameFormat,
   CIM.t$System.PrimaryOwnerContact,
   CIM.t$System.PrimaryOwnerName
FROM
   CIM.t$ManagedSystemElement,
   CIM.t$System
WHERE
   CIM.t$ManagedSystemElement.id$ = CIM.t$System.id$;
```

In addition to the properties of the class, the view has the following two additional fields:

- **Id$:** An object identifier that uniquely identifies the particular instance of the class. See “Object Identifier Id$” on page 425.
- **Class$:** An integer field that identifies the actual type of the class. For example, the actual type of a CIM_System can be any of the concrete subclasses of CIM_System.

Views can be queried using the SELECT statement and updated using the UPDATE statement. Because views cannot be used with the INSERT and DELETE statements, use the constructor and destructor procedures.

Object Identifier Id$

Id$ is a 64-bit object identifier that uniquely identifies a particular instance of a class. This object identifier is usually used as an opaque handle to a particular instance. Id$ is modeled as a signed number for ease of manipulation in Java as a long data type.

Id$ contains the following three parts of information, which can each be extracted by invoking the appropriate stored procedure.

- The most significant 16 bits of id$ encode the actual class of the object.

  This field can be extracted using the MW_DBA.extractClass() function. This field is used for type decisions or to access additional information about the class from the MW_DBA.Class table.

- The next 8 bits of id$ encode the site ID.

  The site ID uniquely identifies the database on a particular site. This field makes the object identifier unique across as many as 256 sites so that inventory data from multiple sites can be rolled up into a single database (Root Server with database) for querying and reporting without causing key conflicts. The site ID can be extracted using the MW_DBA.extractSite() function.

- The least significant 40 bits uniquely identify the particular instance of that class.

  This part can be extracted using the MW_DBA.extractId() function. This is not useful from an end-user's perspective.

The id$ field is used in its entirety as an opaque handle to an instance of a class. When an association class represents a relationship between instances of two classes, the reference fields of the association hold the id$ of the referenced instances (like the pointers). Therefore, id$ and these reference fields are frequently used in Join conditions when constructing the database queries that reference more than one view.

Constructor

Each concrete (non-abstract) CIM class has a constructor-stored-procedure that must be called to create an instance of the class. This stored procedure has input parameters that allow the user to specify a value for each property in the class, and a single output parameter that returns the id$ allocated to the created instance. The application uses this returned id$ value to construct association classes that reference that particular instance.

The constructor is named by prefixing the root name with c$, and each parameter is named by prefixing the root property name with p$. For example, the constructor for CIM_UnitaryComputerSystem, a subclass of CIM_System, is named CIM.c$UnitaryComputerSystem and is constructed for Oracle as shown in the following example.

```
CREATE PROCEDURE CIM.c$UnitaryComputerSystem
```

```
(

p$id$  OUT NUMBER,

p$Caption IN CIM.t$ManagedSystemElement.Caption%TYPE DEFAULT
NULL,

p$Description IN CIM.t$ManagedSystemDescription%TYPE DEFAULT
NULL,

p$InstallDate IN CIM.t$ManagedSystemElement.InstallDate%TYPE
DEFAULT NULL,

p$Status IN CIM.t$ManagedSystemElement.Status%TYPE DEFAULT
NULL,

p$CreationClassName IN CIM.t$System.CreationClassName%TYPE
DEFAULT NULL,

p$Name IN CIM.t$ManagedSystemElement.Name%TYPE DEFAULT NULL,

p$PrimaryOwnerContact IN
CIM.t$System.PrimaryOwnerContact%TYPE DEFAULT NULL,

p$PrimaryOwnerName IN CIM.t$System.PrimaryOwnerName%TYPE
DEFAULT NULL,

p$NameFormat IN CIM.t$System.NameFormat%TYPE DEFAULT NULL,

p$LastLoadInfo IN
CIM.t$UnitaryComputerSystem.LastLoadInfo%TYPE DEFAULT NULL,

p$ResetCapability IN
CIM.t$UnitaryComputerSystem.ResetCapability%TYPE DEFAULT
NULL,

p$PowerManagementSupported IN
CIM.t$UnitaryComputerSystem.PowerManagementSupported%TYPE
DEFAULT NULL,

p$PowerState IN CIM.t$UnitaryComputerSystem.PowerState%TYPE
DEFAULT NULL

)IS

  temp NUMBER;

BEGIN

  LOOP

  SELECT CIM.s$UnitaryComputerSystem.NEXTVAL INTO temp FROM
DUAL;
```

```
  SELECT MW_DBA.makeId(240, temp) INTO temp FROM DUAL;

  EXIT WHEN MOD(temp,100) != 0;

  END LOOP;

  p$id$ := temp;

INSERT INTO CIM.t$ManagedSystemElement (id$, classOid$,
Caption, Description, InstallDate, Status, Name)VALUES(
p$id$, HEXTORAW('0302100203'), p$Caption, p$Description,
p$InstallDate, p$Status, p$Name);

INSERT INTO CIM.t$System ( id$, CreationClassName,
PrimaryOwnerContact, PrimaryOwnerName, NameFormat )VALUES(
p$id$, p$CreationClassName, p$PrimaryOwnerContact,
p$PrimaryOwnerName, p$NameFormat );

INSERT INTO CIM.t$UnitaryComputerSystem (id$, LastLoadInfo,
ResetCapability, PowerManagementSupported, PowerState )
VALUES( p$id$, p$LastLoadInfo, p$ResetCapability,
p$PowerManagementSupported, p$PowerState );

END;
```

Stored procedures can be called with either positional arguments or keyword arguments, or with a combination of the two. If any positional arguments are supplied, they must precede any keyword arguments. Always use keyword arguments when calling constructor Stored Procedures. This provides better insulation from CIM schema changes that cause either the insertion of extra parameters or the recording of existing parameters, either of which can break a positional call in a possible undetectable way. The procedures are generated such that any omitted parameters will default to NULL.

It is permissible to use the positional notation for the first parameter pid, which is the output parameter that returns the object identifier of the newly created instance.

The following code sample shows how to call a stored procedure using positional notation for the first argument and keyword notation for all subsequent arguments on Sybase.

```
CallableStatement CS =

conn.prepareCall( "{call CIM.c$UnitaryComputerSystem( ?,
p$Name=?, p$Description=?)}" )

cs.registerOutParameter ( 1, java.sql.Types.BIGINT ); //id$

cs.setString( 2, "Bogus_UCS_1") ; //Name
```

```
cs.setString( 3, "Created with mixture of positional & keyword
args" ); // Description

cs.executeUpdate();

long id = cs.getLong ( 1 );

SQLWarning w = cs.getWarnings();

if( w != null )

  printWarnings( w );

else

  System.out.println("Created UCS id$ = " + id );
```

The syntax for keyword notation differs in Sybase ASA and Oracle. In Sybase ASA, the syntax is KEYWORD=*value*. In Oracle, the syntax is KEYWORD => *value*. Properly written code will dynamically construct the call string using syntax appropriate for the database in use.

Destructor

Each non-abstract CIM class has a destructor stored procedure that is called to destroy an instance of the class. This stored procedure has only one input parameter that specifies the object identifier (id$) of the instance to be destroyed and returns no value.

The destructor deletes the appropriate rows in all relevant tables, including the rows in the inheritance chain and any associations that reference the instance being destroyed. Only the association is destroyed; the associated objects associated are not destroyed. If there is need to destroy the association, the programmers must ensure that they are not destroyed.

The destructor is named by prefixing the root name with d$ and the single object identifier parameter is named pid. This procedure is called using positional notation. For example, the destructor for CIM_UnitaryComputerSystem, a concrete subclass of CIM_System, is named as CIM.d$UnitaryComputerSystem.

Physical Schema

The physical schema comprises elements necessary to implement the database. The physical schema differs for each database. A typical physical schema consists of:

- Table definitions 't$*xxx*'

- Index definitions 'i$*xxx*'
- Trigger definitions 'x$*xxx*', 'n$*xxx*' and 'u$*xxx*'
- Sequence definitions (Oracle) 's$*xxx*'
- Stored procedures and functions

The logical schema is layered on top of the physical schema and makes it unnecessary for users and applications to know the physical schema.

Inventory Database Schema in ZfD3

The following section describes the database schema classes and the extensions and associations made to the CIM schema for use in ZfD. These extensions have ZENworks or ManageWise as their schema name. ZENworks.*classname* refers to the extended class in the ZENworks schema and ManageWise.*classname* refers to the extended class in the ManageWise schema.

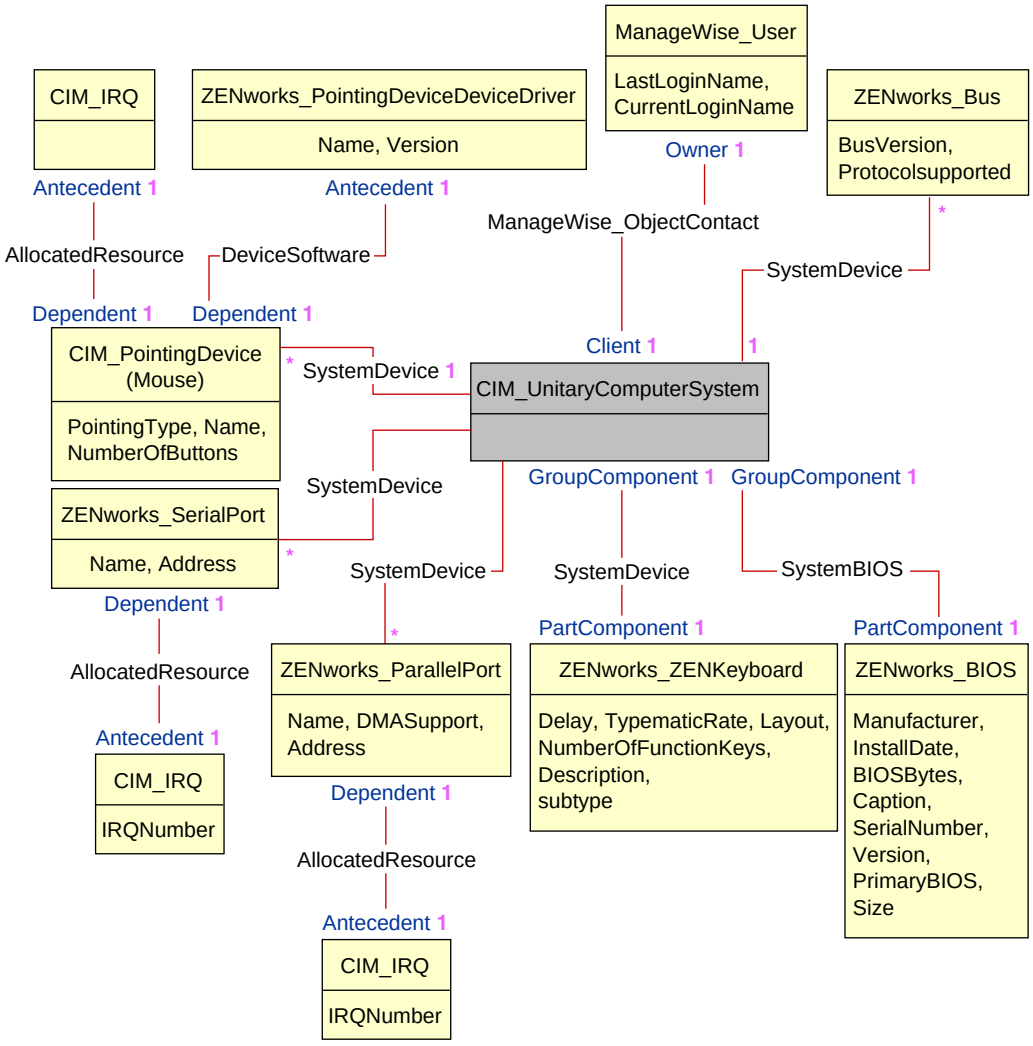
The following sections will help you understand the ZfD database schema:

- "Case Study of CIM Schema Implementation in ZfD3" on page 430
- "Legends for Schema Diagrams" on page 432
- "Leaf Objects in the Database" on page 432
- "Non-CIM Tables and Views in the ZfD Database Schema" on page 433
- "CIM Classes and Extension Classes in ZfD" on page 434
- "Schema Diagrams of CIM and the Extension Schema in ZfD" on page 438
- "Sample Inventory Database Queries" on page 445

Case Study of CIM Schema Implementation in ZfD3

The following scenario describes a managed workstation that has two parallel ports with a specified interrupt number.

In the following schema diagram, the CIM_UnitaryComputerSystem represents a managed inventory system and is shown shaded in the following figure:



The schema diagram illustrates the following:

- All components that a computer system manages are represented as associations from the UnitaryComputerSystem class. The type of references (1..n, 1..1) between two classes are marked.
- Those associations that do not have a schema name are assumed as CIM schema.

  There are three instances of ZENworks_ParallelPort associated to one instance of: CIM_UnitaryComputerSystem using three instances of

CIM_SystemDevice associations, CIM_SystemDevice.GroupComponent references UnitaryComputerSystem, CIM_SystemDevice.PartComponent references ParallelPort

This is called 1 to n object reference relationship and is depicted in the illustration as 1..*. Similarly every instance of ParallelPort has a corresponding instance of CIM_IRQ designating the port's irq. This is one-to-one relationship and is depicted as 1..1.

All other classes follow similar representation. See "CIM Classes and Extension Classes in ZfD" on page 434 for an explanation of the CIM and extended classes. See "Schema Diagrams of CIM and the Extension Schema in ZfD" on page 438 for schema diagrams of other classes.

Legends for Schema Diagrams

The legends for reading the schema diagrams are as follows:

- Class names are enclosed in boxes with the class name as the heading and the attribute names within it.
- Red lines connect two classes using an association class.
- The association class name is shown within the line joining two classes.
- References of the association class are marked on either side of the associated classes.

See the CIM 2.2 schema specification on the DMTF Web site for an explanation about CIM schema. See the DMTF Web site (http://www.dmtf.org).

Leaf Objects in the Database

A leaf object in the Inventory database has only one instance at any given point of time. In terms of relational mapping, a table that is made as a leaf object will have only one row and this row will be shared across multiple workstation objects in the database. The advantages of this approach are better optimized storage and better performance for queries.

Any Inventory component object behaves like a leaf object in the database if it is same across several workstations and no other objects are associated under it. For example, a software application installed on the workstation is a leaf object.

The following classes are leaf nodes in the ZfD Inventory database:

1. ZENworks_PointingDeviceDeviceDriver
2. ZENworks_ZENKeyboard
3. ZENworks_Bus
4. CIM_IRQ
5. CIM_DMA
6. ZENworks_ZENOperatingSystem
7. CIM_Processor
8. CIM_Memory
9. CIM_VideoBIOSElement
10. CIM_Product
11. ZENworks_NetwareClient
12. CIM_POTSModem
13. ZENworks_SoundCard
14. ZENworks_MotherBoard
15. CIM_PowerSupply
16. ZENworks_PhysicalDisk
17. ZENworks_PhysicalDiskette
18. ZENworks_LogicalDiskette
19. ZENworks_PhysicalCDROM
20. Zenworks.NetworkAdapterDriver

"CIM Classes and Extension Classes in ZfD" on page 434 describes the CIM classes that ZfD3 uses.

Non-CIM Tables and Views in the ZfD Database Schema

There are a few tables called "support tables" in the ZfD database. These tables and views do not follow the CIM specification. These tables and views are as follows:

View Name	Description	Table that the View Uses	Inventory Component that Uses the View
ZENworks.Processor.Family	Contains the enum values and strings indicating the processor families. These enum values are described in DMI specification.	t$ProcessorFamily	Inventory Reporting
ZENworks.ProcessorRole	Contains the enum values and strings indicating the processor roles. These enum values are described in DMI specification.	t$ProcessorRole	Inventory Reporting
ZENworks.OperatingSystemType	Contains the enum values and strings indicating the types of operating systems. These enum values are described in DMI specification.	t$OperatingSystemType	Inventory Reporting
ZENworks.VideoArchitecture	Contains the enum values and strings indicating the different video adapter types. These enum values are described in DMI specification.	t$VideoArchitecture	Inventory Reporting
MW_DBA.LockTable	Inventory Storer maintains modification time stamps indicating the last modified time of the inventory information. The Storer may not store the inventory scan received if the time stamp is same.	t$LockTable	Inventory Storer

CIM Classes and Extension Classes in ZfD

The following table describes the CIM and extension classes that ZfD uses:

CIM and Extension Class in ZfD	Description of the details that the Class Models
CIM.PointingDevice	Any pointing device available on the managed system. Mostly used to model the mouse.
ZENworks.SystemInfo	Identification details about the system such as serial number, asset tag, and others.
ZENworks.Site	Site ID and site name of the Inventory database.
ZENworks.PointingDeviceDeviceDriver	Device driver that is installed with the pointing device.
ZENworks.SerialPort	Serial ports on the managed system.
ZENworks.ParallelPort	Parallel ports on the managed system.
ZENworks.ZENKeyboard	Attributes modeling the properties of the system keyboard.
ZENworks.BIOS	BIOS software on the system.
ZENworks.Bus	System bus in the system.
ManageWise.User	Details of the user who was logged in to the workstation.
ManageWise.MSDomainName	Name of the domain to which the Windows NT workstation is attached.
ManageWise.NDSName	DN name and tree under which the managed workstation is registered in NDS.
CIM.VideoBIOSElement:	Video driver.
CIM.Processor	Processor of the workstation.
CIM.Memory	Total memory of the workstation.
ZENworks.VirtualMemory	Total virtual memory size of the workstation.

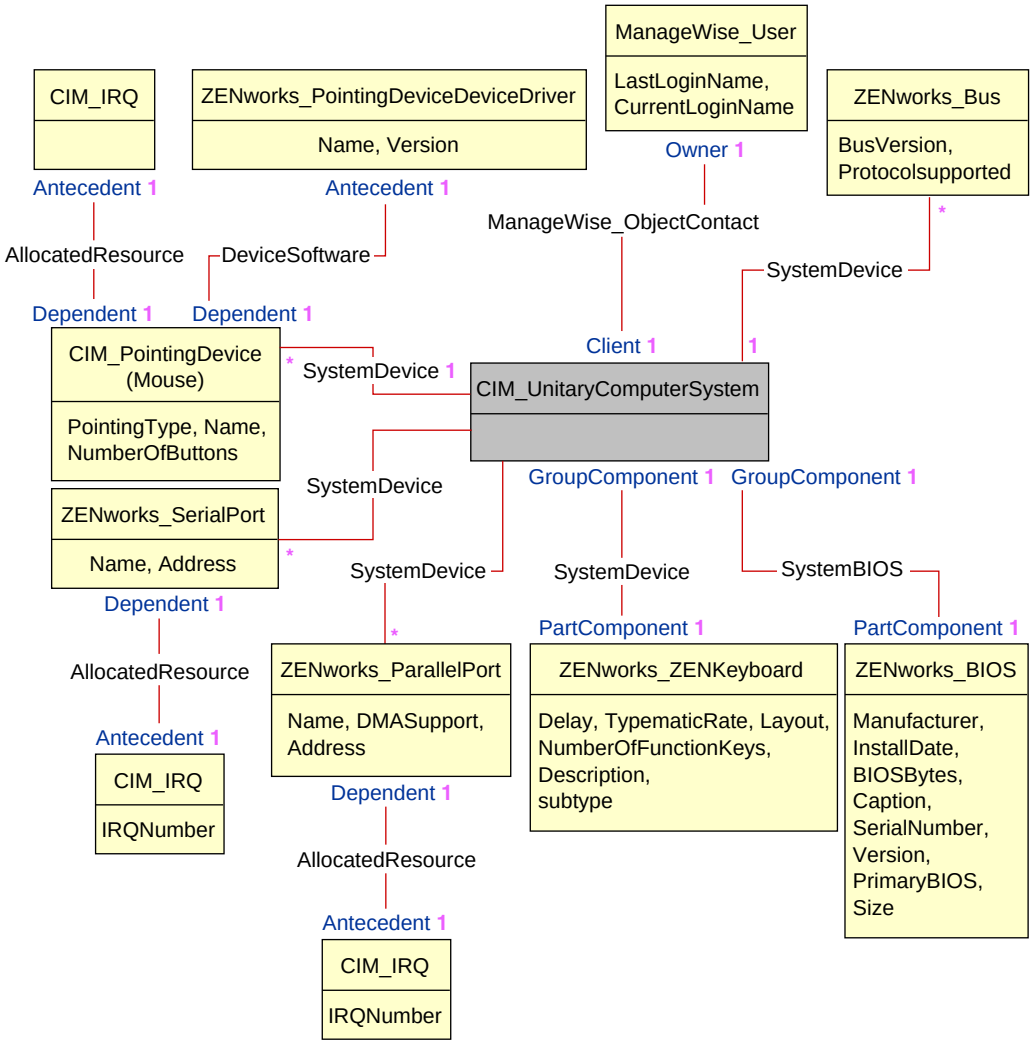
CIM and Extension Class in ZfD	Description of the details that the Class Models
ZENworks.Videoadapter	Properties of the monitor and the adapter connecting it.
ZENworks.ZENOperatingSystem	Details of the operating system.
ZENworks.InventoryScanner	Details of the inventory scanner that has scanned for hardware and software details of the managed workstation
ZENworks.NetwareClient	NetWare client version of the workstation
CIM.Product	Software installed on the managed system. Key attributes are the names of the product, vendor, and version.
CIM.EthernetAdapter	Information on the properties of the network adapter.
ZENworks.NetworkAdapterDriver	Network card adapter driver information.
CIM.IPProtocolEndpoint	IP address of the workstation.
CIM.IPXProtocolEndpoint	IPX™ address of the workstation.
CIM.LANEndpoint	Active MACAddress.
ManageWise.DNSName	DNS Name of the workstation.
ZENworks.SoundCard	Description of the multimedia adapter on the workstation.
CIM_POTSModem	Physical configuration of the modem device.
CIM_DMA	Information about the system DMA channels.
CIM.CacheMemory	Information about the configured system cache.

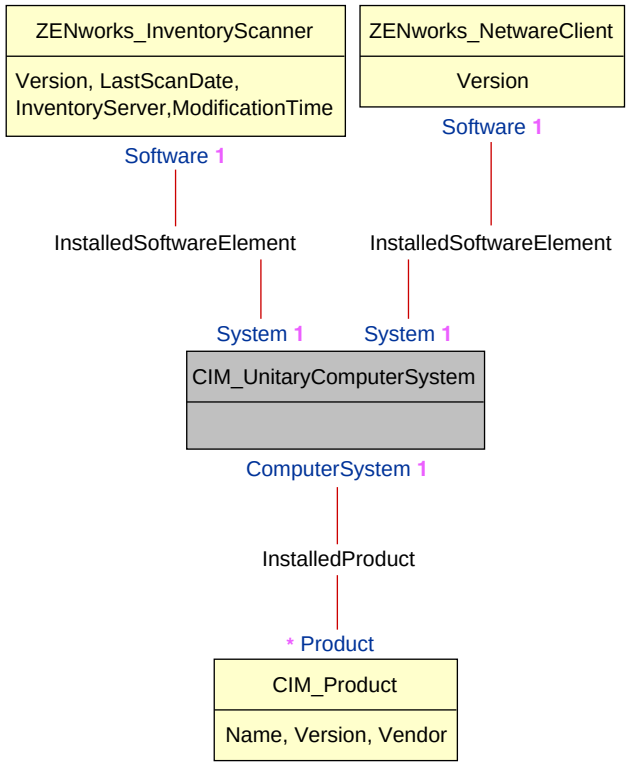
CIM and Extension Class in ZfD	Description of the details that the Class Models
CIM.IRQ	List of Interrupt channels and their status on the system. They are also associated to devices that use the specified interrupt number.
ZENworks.MotherBoard	Information about the motherboard on the workstation.
CIM.PowerSupply	Information about the power supply unit of the workstation.
CIM.Battery	Physical details of the system battery.
CIM.Card	Details of adapter cards mounted on the system board.
CIM.Slot	Expansion slots available on the system board.
ZENworks.StoragePhysicalMedia	Physical information about the storage devices on the workstation, such as hard disk, floppy drives, CD drives, and others.
ZENworks.LogicalDiskette	Drive mapped to the floppy drive.
ZENworks.PhysicalDiskette	Derived from ZENworks.StoragePhysicalMedia to model the floppy disk drive.
ZENworks.PhysicalDiskDrive	Derived from ZENworks.StoragePhysicalMedia to model the hard disk.
ZENworks.LogicalDiskDrive	Information about the local drives on the hard disk.
CIM.LocalFileSystem	Information about the local file system mounted on the hard disk.
ZENworks.PhysicalCDROM	Derived from ZENworks.StoragePhysicalMedia to model the CD drive.
ZENworks.LogicalCDROM	Drive mapped to the CD drive.

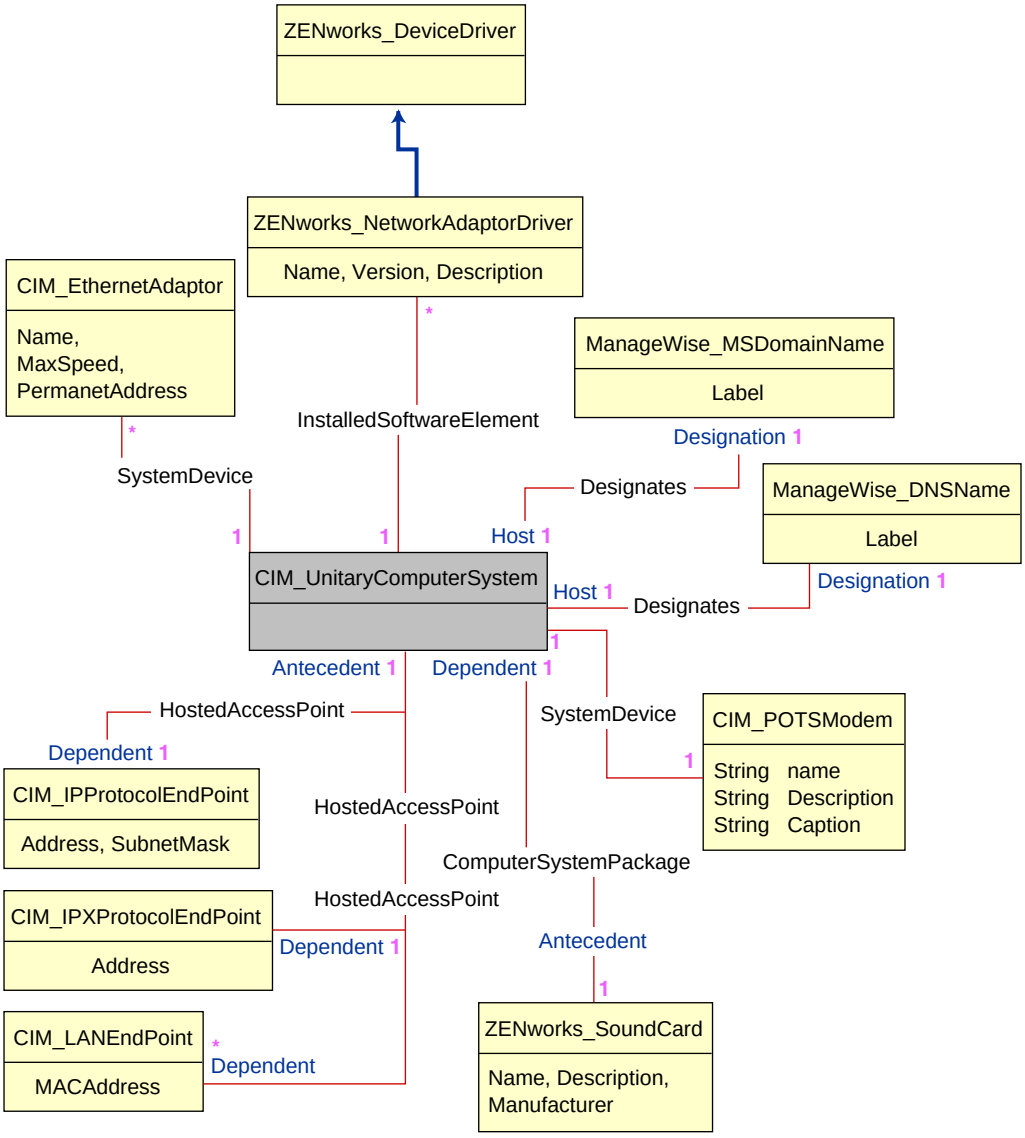
CIM and Extension Class in ZfD	Description of the details that the Class Models
ZENworks.BackupDisk	Derived from ZENworks.StoragePhysicalMedia to model other backup disks like Jaz drives and Zip disks.
ZENworks.LogicalBackupDisk	Drives mapped to the backup devices on the workstation.

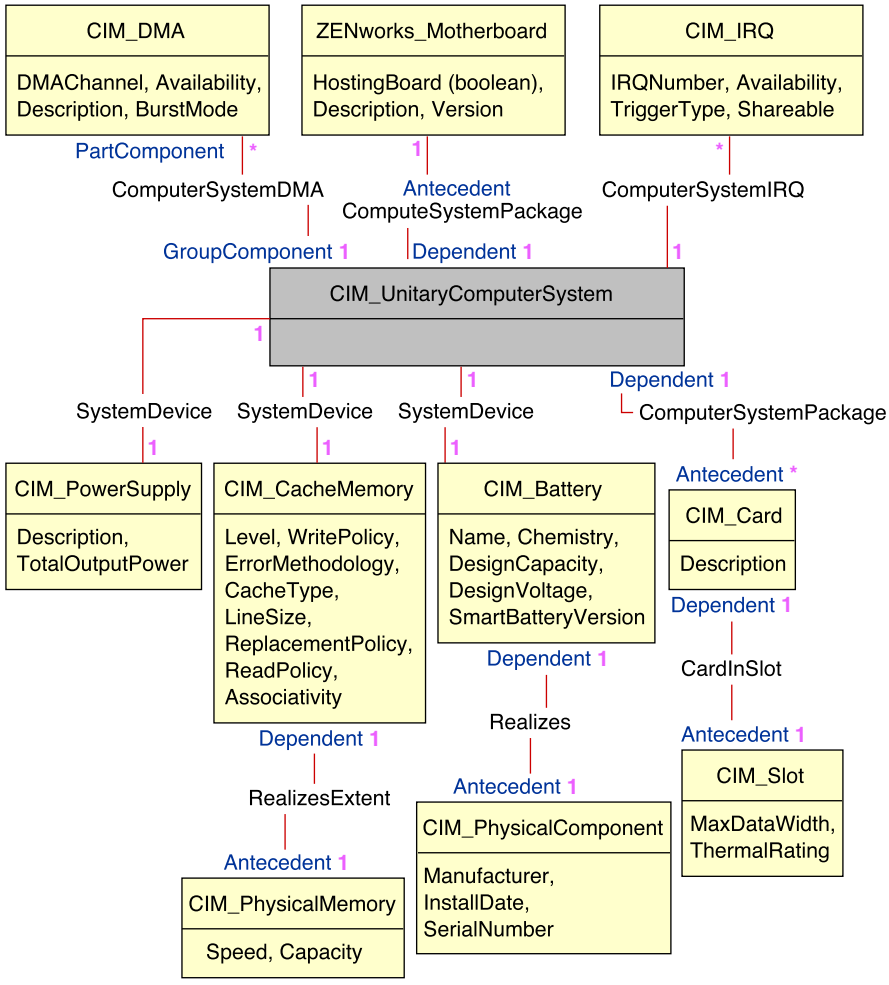
Schema Diagrams of CIM and the Extension Schema in ZfD

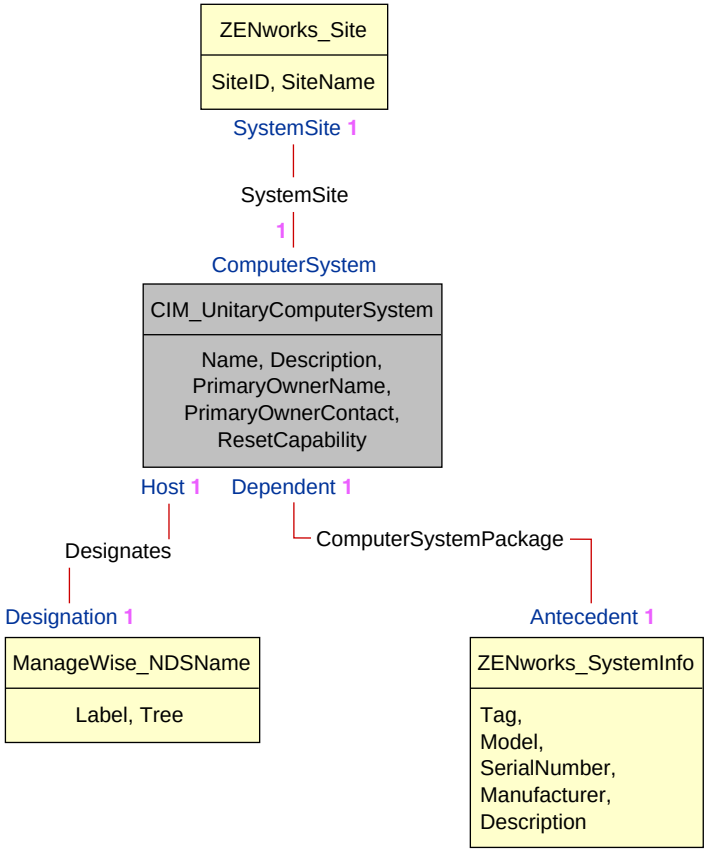
The following schema diagrams of the CIM and extension schema model the Inventory database in ZfD:

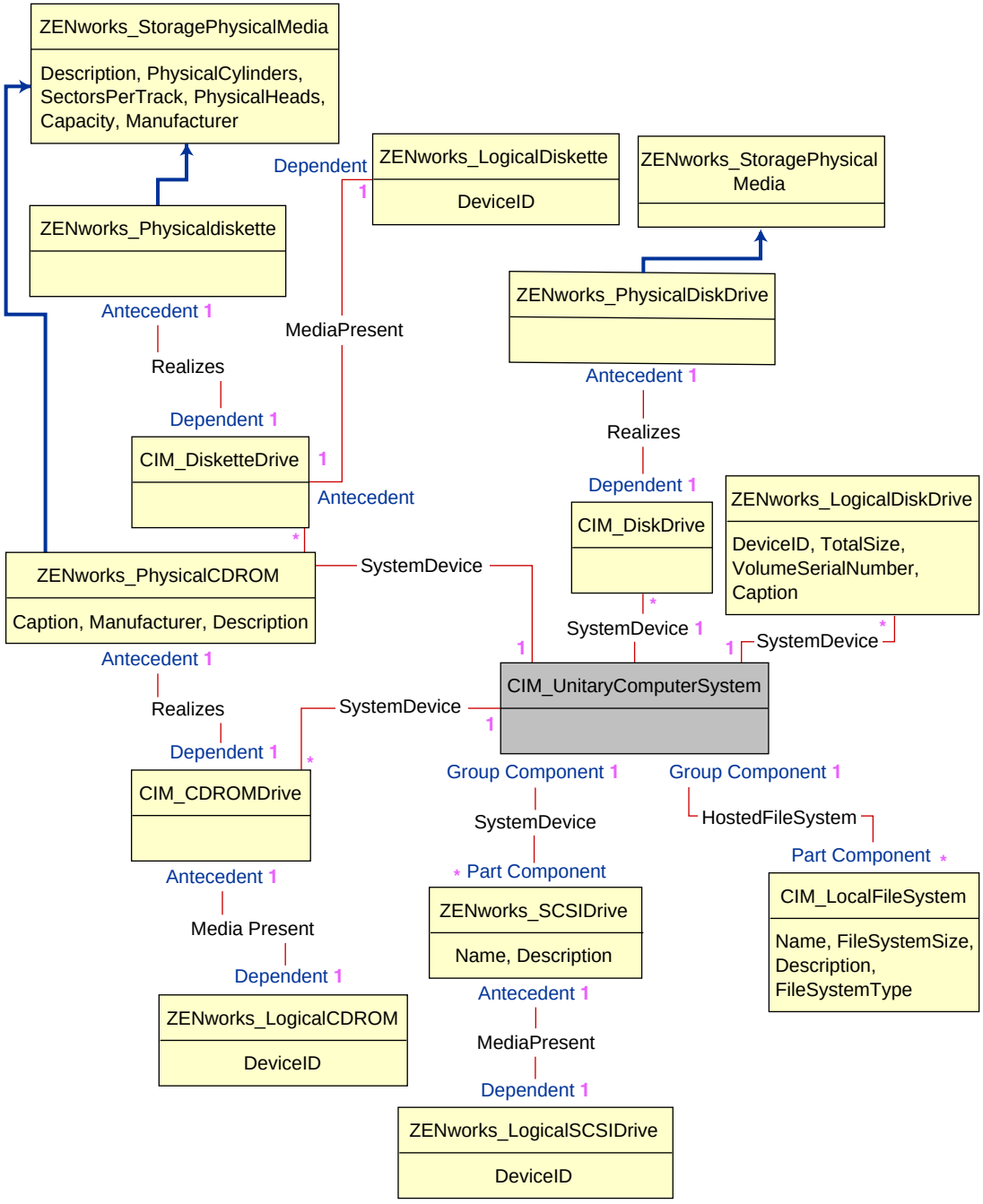

ManageWise_User
LastLoginName, CurrentLoginName
CIM_IRQ
ZENworks_PointingDeviceDeviceDriver
Name, Version
ZENworks_Bus
BusVersion, Protocolsupported
Owner 1
Antecedent 1
Antecedent 1
ManageWise_ObjectContact
*
AllocatedResource
DeviceSoftware
SystemDevice
Dependent 1
Dependent 1
Client 1
1
CIM_PointingDevice (Mouse)
PointingType, Name, NumberOfButtons
*
SystemDevice 1
CIM_UnitaryComputerSystem
SystemDevice
GroupComponent 1
GroupComponent 1
ZENworks_SerialPort
Name, Address
*
SystemDevice
SystemDevice
SystemBIOS
Dependent 1
*
PartComponent 1
PartComponent 1
AllocatedResource
ZENworks_ParallelPort
Name, DMASupport, Address
ZENworks_ZENKeyboard
Delay, TypematicRate, Layout, NumberOfFunctionKeys, Description, subtype
ZENworks_BIOS
Manufacturer, InstallDate, BIOSBytes, Caption, SerialNumber, Version, PrimaryBIOS, Size
Antecedent 1
CIM_IRQ
IRQNumber
Dependent 1
AllocatedResource
Antecedent 1
CIM_IRQ
IRQNumber


ZENworks_InventoryScanner
Version, LastScanDate, InventoryServer,ModificationTime
ZENworks_NetwareClient
Version
Software 1
Software 1
InstalledSoftwareElement
InstalledSoftwareElement
System 1
System 1
CIM_UnitaryComputerSystem
ComputerSystem 1
InstalledProduct
* Product
CIM_Product
Name, Version, Vendor


ZENworks_DeviceDriver
ZENworks_NetworkAdaptorDriver
Name, Version, Description
CIM_EthernetAdaptor
Name,
MaxSpeed,
PermanetAddress
ManageWise_MSDomainName
Label
InstalledSoftwareElement
Designation 1
SystemDevice
Designates
ManageWise_DNSName
Label
Host 1
CIM_UnitaryComputerSystem
Host 1
Designates
Designation 1
Antecedent 1
Dependent 1
HostedAccessPoint
SystemDevice
CIM_POTSModem
String name
String Description
String Caption
Dependent 1
CIM_IPProtocolEndPoint
Address, SubnetMask
HostedAccessPoint
ComputerSystemPackage
CIM_IPXProtocolEndPoint
Address
HostedAccessPoint
Dependent 1
Antecedent
CIM_LANEndPoint
MACAddress
Dependent
ZENworks_SoundCard
Name, Description,
Manufacturer


CIM_DMA
DMAChannel, Availability, Description, BurstMode
ZENworks_Motherboard
HostingBoard (boolean), Description, Version
CIM_IRQ
IRQNumber, Availability, TriggerType, Shareable
PartComponent
*
1
*
ComputerSystemDMA
Antecedent
ComputeSystemPackage
ComputerSystemIRQ
GroupComponent 1
Dependent 1
1
CIM_UnitaryComputerSystem
1
1
1
Dependent 1
SystemDevice
SystemDevice
SystemDevice
ComputerSystemPackage
1
1
1
CIM_PowerSupply
Description, TotalOutputPower
CIM_CacheMemory
Level, WritePolicy, ErrorMethodology, CacheType, LineSize, ReplacementPolicy, ReadPolicy, Associativity
CIM_Battery
Name, Chemistry, DesignCapacity, DesignVoltage, SmartBatteryVersion
Antecedent *
CIM_Card
Description
Dependent 1
Dependent 1
CardInSlot
Realizes
Dependent 1
Antecedent 1
CIM_Slot
MaxDataWidth, ThermalRating
RealizesExtent
Antecedent 1
CIM_PhysicalComponent
Manufacturer, InstallDate, SerialNumber
Antecedent 1
CIM_PhysicalMemory
Speed, Capacity


ZENworks_Site
SiteID, SiteName
SystemSite 1
SystemSite
1
ComputerSystem
CIM_UnitaryComputerSystem
Name, Description,
PrimaryOwnerName,
PrimaryOwnerContact,
ResetCapability
Host 1
Dependent 1
Designates
ComputerSystemPackage
Designation 1
Antecedent 1
ManageWise_NDSName
Label, Tree
ZENworks_SystemInfo
Tag,
Model,
SerialNumber,
Manufacturer,
Description


ZENworks_StoragePhysicalMedia
Description, PhysicalCylinders, SectorsPerTrack, PhysicalHeads, Capacity, Manufacturer
Dependent
ZENworks_LogicalDiskette
DeviceID
ZENworks_StoragePhysical Media
ZENworks_Physicaldiskette
Antecedent 1
MediaPresent
ZENworks_PhysicalDiskDrive
Realizes
Dependent 1
Antecedent 1
CIM_DisketteDrive
Antecedent
Realizes
Dependent 1
ZENworks_LogicalDiskDrive
DeviceID, TotalSize, VolumeSerialNumber, Caption
CIM_DiskDrive
ZENworks_PhysicalCDROM
Caption, Manufacturer, Description
SystemDevice
SystemDevice 1
SystemDevice
Antecedent 1
CIM_UnitaryComputerSystem
Realizes
SystemDevice
Dependent 1
Group Component 1
Group Component 1
CIM_CDROMDrive
SystemDevice
HostedFileSystem
Part Component
Antecedent 1
Part Component
CIM_LocalFileSystem
Name, FileSystemSize, Description, FileSystemType
Media Present
ZENworks_SCSIDrive
Name, Description
Dependent 1
Antecedent 1
ZENworks_LogicalCDROM
DeviceID
MediaPresent
Dependent 1
ZENworks_LogicalSCSIDrive
DeviceID

Sample Inventory Database Queries

The following are sample queries for retrieving the inventory information from the ZfD Inventory database.

Refer to the schema diagrams in "Schema Diagrams of CIM and the Extension Schema in ZfD" on page 438 to find out the associated schema classes and attribute information.

1. Retrieve the name and ID of all workstations from the database and also to the tree to which these workstations are registered. The query is as follows:

```
SELECT u.id$,m.label,m.tree FROM managewise.NDSName
m,cim.UnitaryComputerSystem u,managewise.Designates s
where s.Designation=m.id$and s.HOST=u.id$
```

2. Retrieve the asset tag, manufacturer, and serial number of all the workstations in the database. The query is as follows:

```
SELECT m.Tag,m.Manufacturer,m.SerialNumber FROM
cim.UnitaryComputerSystem u,zenworks.SystemInfo
m,cim.ComputerSystemPackage s WHERE s.Antecedent=m.id$and
s.Dependent=u.id$
```

3. Retrieve all the software applications with their versions that are installed on the workstation 'SJOHN164_99_139_79.WS' registered under 'NOVELL_AUS' tree. The query is as follows:

```
SELECT m.name,m.version FROM cim.Product
m,cim.UnitaryComputerSystem u,zenworks.InstalledProduct
s,managewise.NDSName m1,managewise.Designates s1 WHERE
(s.Product=m.id$and s.ComputerSystem=u.id$) AND
(s1.Designation=m1.id$and s1.Host=u.id$) AND
m1.label='SJOHN164_99_139_79.WS' and m1.tree='Novell_AUS'
```

4. Retrieve the processor information for the workstation 'SJOHN164_99_139_79.WS'. The query is as follows:

```
SELECT
m.DeviceID,m.Family,m.Stepping,m.OtherFamilyDescription,
m.MaxClockSpeed,m.Role,m.UpgradeMethod FROM cim.Processor
m,cim.UnitaryComputerSystem
u,cim.ComputerSystemProcessor s managewise.NDSName
m1,managewise.Designates s1 WHERE
(s.PartComponent=m.id$and s.GroupComponent=u.id$) AND
m1.label='SJOHN164_99_139_79.WS'
```

5. Retrieve the ID of the UnitaryComputerSystem used for the workstation 'SJOHN164_99_139_79.WS'. The query is as follows:

```
SELECT s.host FROM managewise.NDSName
m,managewise.Designates s WHERE
m.label='SJOHN164_99_139_79.WS' and m.id$=s.Designation
```

6. Having known the ID of the UnitaryComputerSystem for a particular workstation from the query as shown in query 5, query 4 can be modified as:

```
SELECT
m.DeviceID,m.Family,m.Stepping,m.OtherFamilyDescription,
m.MaxClockSpeed,m.Role,m.UpgradeMethod FROM cim.Processor
m,cim.UnitaryComputerSystem u,
cim.ComputerSystemProcessor s u.id$=? and
s.PartComponent=m.id$ and s.GroupComponent=u.id$
```

   Substitute the ID of the specified workstation in place of the ? value for u.id in the query.

7. List the IP Address, IPX Address, and MACAddress of all workstations in the database. The query is as follows:

```
SELECT ip.Address, ipx.Address, mac.MACAddress FROM
cim.IPProtocolEndpoint ip, cim.IPXProtocolEndpoint ipx,
cim.LANEndpoint mac, cim.UnitaryComputerSystem u,
cim.HostedAccessPoint s WHERE (s.Dependent=ip.id$ and
s.Antecedent=u.id$) AND (s.Dependent=ipx.id$ and
s.Antecedent=u.id$) AND (s.Dependent=mac.id$ and
s.Antecedent=u.id$)
```

   Modify the same query to get the information for a specified workstation as follows:

```
SELECT ip.Address, ipx.Address, mac.MACAddress FROM
cim.IPProtocolEndpoint ip, cim.IPXProtocolEndpoint ipx,
cim.LANEndpoint mac, cim.UnitaryComputerSystem u,
cim.HostedAccessPoint s WHERE (s.Dependent=ip.id$ and
s.Antecedent=u.id$) AND (s.Dependent=ipx.id$ and
s.Antecedent=u.id$) AND (s.Dependent=mac.id$ and
s.Antecedent=u.id$)AND u.id$=?
```

   Use the query as shown in query 5 to retrieve the ID of the specified workstation and substitute the ID in place of the ? value for u.id in the query.

8. Retrieve the name and other properties of the drives on the hard disk of the specified workstation.

```
SELECT m.DEVICEID, m.TotalSize, m.VolumeSerialNumber,
m.Caption FROM zenworks.LogicalDiskDrive m,
cim.UnitaryComputerSystem u, cim.SystemDevice s WHERE
s.PartComponent=m.id$ AND s.GroupComponent=u.id$ and
u.id$=?
```

Use the query shown in query 5 to retrieve the ID of the specified workstation and substitute the ID in place of the ? for u.id$ in the query.

23 Setting Up Workstation Inventory

The following sections contain detailed information to help you set up ZENworks™ for Desktops (ZfD) Workstation Inventory:

- "Configuring the Settings for the Inventory Service Object" on page 449
- "Configuring the Inventory Settings for the Workstation Object" on page 451
- "Configuring the Workstation Inventory Policy Settings" on page 452
- "Configuring the Roll-Up Policy Settings" on page 453
- "Customizing the Hardware Scanning Information of Jaz, Zip, and Floppy Drive Vendors" on page 456
- "Scanning for IBM Computer Models" on page 454
- "Customizing Software Scanning of Workstations" on page 457
- "Customizing the Software Scanning Information of Vendors and Products" on page 459
- "Changing the Role of the Inventory Service Object" on page 461
- "Assigning Trustees to the Database Object" on page 476
- "Migrating ZENworks 2 Inventory Information" on page 476
- "Setting up the Inventory Database" on page 484

Configuring the Settings for the Inventory Service Object

The Inventory Service object settings configure the scanning for the associated workstations. From the Inventory Service Object property page, you can configure the following:

- Inventory Server Role
- Discard Scan Data Files
- Scan Directory Path
- Enable scan of workstations
- Start full scan

To open the Inventory Service Object properties page:

1. In ConsoleOne™, right-click the Inventory Service object (*servername*_ZenInvservice) > click Properties > click the Inventory Service Object Properties tab.
2. Modify the following settings:

   Inventory Server Role: Based on the servers that you have deployed for scanning inventory, you must specify the role of the server. See "Changing the Role of the Inventory Service Object" on page 461.

   Discard Scan Data Time: Any scan data files (.ZIP files) that have scan information collected before the Discard Scan Data Time that you specify in the Inventory Service Object Property page will be discarded. The scan data files are removed from the server, which is one of the following types: Intermediate Server, Intermediate Server with Database, Intermediate Server with Database & Workstations, and Intermediate Server with Workstations.

   Scan Directory Path: When you install ZfD, you specify the volume on the server for storing the scan data files. If required, you can modify the volume or the directory of the Scan Directory (SCANDIR) setting in the Inventory Service Object property page. The SCANDIR directory path is the location on the server that stores the scan data files. The format of the Scan Directory Path is as follows:
   server_name\volume_of_the_server_directory.

   You cannot modify the server name specified in the SCANDIR path. If you modify the directory, the directory must already exist.

   To modify the path, click the Browse button.

   Enable Scan of Workstations: To scan the workstations associated with the Inventory Service object, you must enable the scan option listed in the Inventory Service Object property page. To disable the scanning of workstations, uncheck this option.

   Start Full Scan: When scanning the workstation for the first time, the Scanner collects the complete inventory of the workstation. A complete inventory scan of the workstation is referred as a *full scan.* After the workstation is scanned, the next time the Scanner compares the current inventory data to the history data that it maintains. If there are any changes to the workstation, the Scanner creates a *delta scan*, which collects the changes in inventory since the last scan was done. The delta scan setting is the default scan operation for each successive scan after the first scanning of the workstation. If the Status Log reported by the

inventory component indicates the scanning on the workstation is not successful, you can enforce a full Scan. This policy settings is applicable for all workstations associated with it. To override this policy, you set this option for an individual workstation. See Enforce a Full Scan on the Workstation for the Workstation object settings.

3 Click OK.

NOTE: If you are modifying the Inventory policies or configuring the objects, always stop the Inventory Services. Configure the policies and properties of the objects. Restart the Inventory services again.

Configuring the Inventory Settings for the Workstation Object

You use the inventory settings of the Workstation object to identify the .STR filename of the last scan. Identifying the scan data filename is useful to troubleshoot inventory scanning problems for the workstation. See View the Scan Data File for more information.

You configure the inventory settings for the selected workstation by enforcing a complete scan of the workstation using the Workstation Scan Configuration property page. See Enable Full Scan for more information about the Full Scan option.

To configure the inventory settings for the workstation object:

1 In ConsoleOne, right-click the Workstation object > click Properties > ZENworks Inventory > Workstation Scan Configuration properties tab.

2 Check Start Full Scan if required > modify the following settings:

View the Scan Data Filename: The Scanner stores the scan data of the workstation as a scan data file (.STR) in the scan directory (SCANDIR) on the inventory server. The Workstation Scan Configuration properties page displays the scan filename of the last scan done at the workstation. The .STR file uses the following filename conventions: *macAddress_gmt_sequencenumber*.STR where *macAddress* is the MacAddress of the workstation, *gmt* is the time that the workstation was first scanned, *sequencenumber* is the internal sequencing number generated by the Scanner for the workstation while scanning, and .STR is the file extension.

For example, 00508b12b2c4_944029836000_10.STR is the .STR file for the workstation with the MACAddress of 00508b12b2c4, the GMT of the Workstation as 944029836000, and the internal sequencing number of 10.

Start Full Scan: When scanning the workstation for the first time, the scanner collects the complete inventory. A complete inventory scan of the workstation is referred to as a *full scan.* After the workstation is scanned, the next time the scanner compares the current inventory data to the history data that it maintains. If there are any changes to the workstation, the scanner creates a *delta scan*, which collects the changes in inventory since the last scan was done. The delta scan setting is the default scan operation for each successive scan after the first scanning of the workstation. If the status log reported by the inventory component indicates that the scanning on the workstation was not successful, you can enforce a full scan.

3. Click OK.

Configuring the Workstation Inventory Policy Settings

You can set policies to control how workstations collect inventory. The Inventory policy settings configure the inventory scanning options for the selected Workstation Inventory Package. The Inventory policy settings stored in NDS are associated with a Workstation object. Each Workstation object has an associated Inventory policy package.

From the Inventory policy, you can configure the following:

- DN of the Inventory Service Object
- Enable Software Scan
- Software Applications in the Custom Scan Editor

To set a Workstation Inventory policy:

1. Create or modify a workstation policy package.
2. Enable the Workstation Inventory policy.
3. Set the details in the Workstation Inventory policy.

You can complete these steps on your own or use the Policy Wizard to guide you through the process.

To configure the Workstation Inventory policy:

1. In ConsoleOne, right-click the Workstation Inventory policy package > click Properties > click Policies > click one of the following subtabs: Win95-98 or WinNT-2000.
2. Click the Workstation Inventory row > Properties > Workstation Inventory Policy tab.
3. Modify the settings.

   Inventory Service Object: The installation program creates the Inventory Service object and copies the inventory components on the server. In this policy, you choose an Inventory Service object to which the scanners will send the inventory information of the workstations associated with this server.

   Enable Software Scan: By default, the scanners collect hardware information of the workstations. When you check the Enable Software Scan option, the next time the workstations associated with this policy package are scanned, the scanners will collect software information. See "Software Information Collected by the Scanners" on page 357 for more information.

   Custom Scan Editor: If you enable the software scanning of the workstations, specify the list of software applications that will be scanned for at the workstation in the Custom Scan Editor. See "Customizing Software Scanning of Workstations" on page 457 for more information.
4. Click OK.

HINT: If you are modifying the Inventory policies or configuring the objects, always stop the Inventory Services. Configure the policies and properties of the objects. Restart the Inventory services again.

Configuring the Roll-Up Policy Settings

The Roll-Up policy settings configure the selected server for roll-up of scan information. The settings in the Roll-Up policy identifies the next-level server (DN of the Inventory Service object) for moving the scan data from the selected server. These settings stored in NDS are associated with the server object.

To set up the Roll-Up policy:

1. Create or modify a server package.
2. Enable the Roll-Up policy.

3. Configure the Roll-Up policy.

To open the Roll-Up policy:

1. In ConsoleOne, right-click the Server package > click Properties > Policies > click one of the following subtabs: General, NetWare, WinNT-2000.
2. Check the check box under the Enabled column for the zeninvRollup Policy.
3. Click Properties.
4. Modify the setting.

   Destination Server Object: You must specify the DN of the Inventory Service object at the next level for moving the scan data from the selected server. The server that you specify must be another Intermediate Server, Intermediate Server with Database, Intermediate Server with Database and Workstations, Intermediate Server with Workstations, Root Server, or Root Server with Workstations.

   HINT: Ensure that the specified server is a different server because the roll-up of data cannot happen on the same server. Also, the higher level roll-up server cannot specify the lower-level server as the next-destination server for roll-up of data.
5. Click OK.

HINT: If you are modifying the Inventory policies or configuring the objects, always stop the Inventory Services. Configure the policies and properties of the objects. Restart the Inventory services again.

Scanning for IBM Computer Models

The scanner (WINSCAN.EXE) uses the IBMNAMES.INI file to get information about IBM workstations running under Windows 95/98. The scanner reads the model name using the machine type and model number information specified in the IBMNAMES.INI file. This file is located in the SYS:\PUBLIC\ZENWORKS directory of the inventory server.

Using the IBMNAMES.INI file is an IBM-specific method of scanning model names of IBM workstations defined by the scanner. If you have a new IBM workstation that is not listed in the IBMNAMES.INI file, the model number of the workstation will not be scanned. You must edit the IBMNAMES.INI file to add the machine type, model number, and model description of the new

IBM workstation. By adding this entry, you enable the scanner to identify the new model name

Editing the IBMNAMES.INI File to Add the IBM Computer Models

The IBMNAMES.INI file contains the model number and description of IBM* desktops and laptop models. You add entries in this file to specify the models that the scanner should scan for at the workstations.

This file begins with a section called [Product Names].

The format of each entry in the section is as follows:

```
4_bytes_machine_type-3_byte_model_number=model_description
```

For example, if the model is IBM PC 140 and the machine type is 6260, specify the model description as IBM PC 140. The entry in the IBMNAMES.INI is 6260-79T = IBM PC 140.

If you want the scanner to scan for all IBM computer models of a particular machine type with the same model description, the *3_byte_model_number* is specified with three question marks (???) as wildcard characters.

A sample IBMNAMES.INI file is as follows:

```
[Product Names]

6272-???=IBM PC 300GL (6272)

6260-???=IBM PC 140 (6260)

6282-???=IBM PC 300Gl (6282)

2460-79T=IBM ThinkPad 5602 (2641)

...
```

For example, to scan all models of a 6282 machine type with same model description, the file entry is as follows:

```
6282-???=IBM PC 300GL (6282)
```

The machine type and model number are printed at the rear of the laptop or on the backside of the desktop workstation. For example, the 760E Thinkpad model has the following label: TYPE *9546-A98*.

Customizing the Hardware Scanning Information of Jaz, Zip, and Floppy Drive Vendors

The scan information of the vendors for devices such as backup and floppy devices is usually unavailable on the workstation. Also, if the information is available, the vendor information does not usually contain the details. You can customize and update information about the vendors of these devices in the Hardware Rules file. The Scanners read this file during the hardware scanning process for these devices.

The contents of the Hardware Rules file (ZIPPNAMES.INI) is as follows:

```
[Identifier]

device_id=vendor_display_name_you_specify

[Floppy]

device_id=vendor_display_name_you_specify
```

where *device_id* is the unique ID generated and updated in the registry by the vendor during the installation of the device on the workstation.

For example, the contents of the file are as follows:

```
[Identifier]

IOMEGA ZIP 100 D.13=Iomega Corporation
```

This entry is for a 100 MB Zip drive installed on the workstation.

```
[Floppy]

MITBISHI LS-120 F200 08=MITBISHI
```

This entry is for a LS-120 floppy drive installed on the workstation.

To customize and update the vendor information for display:

1. Open the ZIPPNAMES.INI file in a text editor.

   This text file is located in \PUBLIC\ZENWORKS directory on the inventory server.

2. Add or modify the entries in this file.

   If you specify incorrect values for the device id entry, the device will not be displayed in the Inventory windows.

3. Save the changes.

Customizing Software Scanning of Workstations

You can customize the list of software applications that you want to scan for at the managed workstations. You specify the software scan settings in the Workstation Inventory policy page. The software scan settings are saved in NDS®.

By default, the Scanner will not scan for software applications at the workstation. You must enable the Software Scan option in the Workstation Inventory policy and configure the list of applications for scanning.

To specify the applications you want to scan for, you add the list of applications or import files that contain the list of applications. You can also export the list of applications as a file and then modify the file.

If you have a large number of software applications that you want to specify, you can create a Custom Scan file following the conventions explained in this section and later import the file.

To specify software scan settings that you specified at a different location, you export the file at that location and import the file at the location you want to use the list.

The following sections contain more information to help you customize workstation scanning:

- “Adding New Applications for Scanning” on page 457
- “Format of the Custom Scan File” on page 458
- “Exporting the List of Application Files for Scanning” on page 459

Adding New Applications for Scanning

To add a new application, you must provide the details of the application.

To add a new application for scanning:

1. In ConsoleOne, open the Workstation Inventory policy.

   See “Configuring the Workstation Inventory Policy Settings” on page 452 for more information.

   Ensure that the Enable Software Scan option is checked.
2. Click the Custom Scan Editor button.
3. Click Add to specify the details of the application.

4 Fill in the details of the application:

   Vendor name, Product name, Product version, File name, File Size (in Bytes)

5 Click OK.

6 To save the application entry in NDS, click OK in the Custom Scan Editor dialog box.

You can also add application entries to the Custom Scan table by importing a file with the list of application entries. You create this file by following the format of the Custom Scan file conventions. See "Format of the Custom Scan File" on page 458 for more information.

To add a list of new applications:

1 Open a text editor.

2 Create a file with the format specified in "Format of the Custom Scan File" on page 458.

3 Save the application as a text file with any extension you prefer.

4 In ConsoleOne, open the Workstation Inventory policy.

   Ensure that the Enable Software Scan option is checked.

5 Click Custom Scan Editor.

6 Click Import.

   To save the application entry in NDS, click OK in the Custom Scan Editor dialog box.

Format of the Custom Scan File

The contents of the Custom Scan file are as follows:

total_number_of_application_entries_in_Custom_Scan_file;total_number_of_columns_in_the_application_entry

vendor_name;product_name;product_version;file_name;file_size (in Bytes)

vendor_name;product_name;product_version;file_name;file_size (in Bytes)

vendor_name;product_name;product_version;file_name;file_size (in Bytes)

Keep in mind the following guidelines as you work with the Custom Scan file:

- The default total number of columns in the application entry is 5.
- The separator between the columns is a semicolon (;).
- Fill in all the columns for each application entry.
- Do not use comma (,) in the file size parameter.

The following is a sample Custom Scan file:

```
2;5

Novell;GroupWise;5.5;grpwise.exe;4025856

Novell;client32nlm;3.03;client32.nlm;524168
```

Exporting the List of Application Files for Scanning

You can export the Custom Scan file to use at a different location. You export the Custom Scan file at one location and then import it at the other location.

To export the list of applications:

1. In ConsoleOne, open the Workstation Inventory policy.

   See "Configuring the Workstation Inventory Policy Settings" on page 452 for more information.

   Ensure that the Enable Software Scan option is checked.
2. Click Custom Scan Editor.
3. Click Export.
4. Type the filename with any extension for the text file.

   The export file is a text file.
5. Click OK.

The exported file will contain the list of applications that are displayed in the Custom Scan table. If you have not saved the list of applications before exporting, the entries in the exported file and the saved application entries in NDS will differ.

Customizing the Software Scanning Information of Vendors and Products

The software information of the same vendor may sometimes have different vendor names or product names. For example, if the software scan data contains information of more than one product for the same vendor, and if the vendor name differs, the inventory display windows will display the software information under different vendor names.

By default, the software information is displayed for each unique vendor name in the Inventory Query window, Inventory Summary window, and the Inventory reports. If the vendor or product names differ, you can merge the software information.You can also prevent the display of specific vendors and products in the inventory windows. You customize these settings in the Software Rules file. The scanners read the file during the scanning process for displaying the vendor or product name as specified in the file.

The contents of the Software Rules file are as follows:

```
[vendor]
```

scanned_vendor_name_reported_by_scanner=vendor_display_name_you_specify

scanned_vendor_name_reported_by_scanner=vendor_display_name_you_specify

```
[product]
```

scanned_product_name_reported_by_scanner=product_display_name_you_specify

scanned_product_name_reported_by_scanner=product_display_name_you_specify

To customize the vendor and product names for display:

1. Open the SWRULES.INI file in a text editor.

   The SWRULES.INI file is a text file located in \PUBLIC\ZENWORKS on the inventory server.

2. To modify the vendor name, specify the details for *scanned_vendor_name_reported_by_scanner* and the *vendor_display_name_you_specify*.

For example, to display the software vendor information for Novell, Novell Inc., Novell Corp, and Novell Inc as Novell Inc., edit the following section:

```
[vendor]

Novell=Novell Inc.

NOVELL INC=Novell Inc.

NOVELL CORP=Novell Inc.

NOVELL Inc=Novell Inc.
```

3. To modify the product name, specify the Scanned Product Name and the Product Display Name.

   For example, to display the product information: Microsoft(R) Windows (TM) Operating System, Microsoft® Windows, Microsoft(R) Windows NT (R) Operating System as Microsoft, edit the following section.

   ```
   [product]

   Microsoft (R) Windows (TM) Operating System=Microsoft
   Windows

   Microsoft ® Windows=Microsoft Windows

   Microsoft (R) Windows NT (R) Operating System=Microsoft
   Windows
   ```

4. To specify that the scanned information for a product or vendor should not be reported by the scanners, add the following entry in the file:

   ```
   [vendor]

   others=null
   ```

You should follow these rules while editing the SWRULES.INI file:

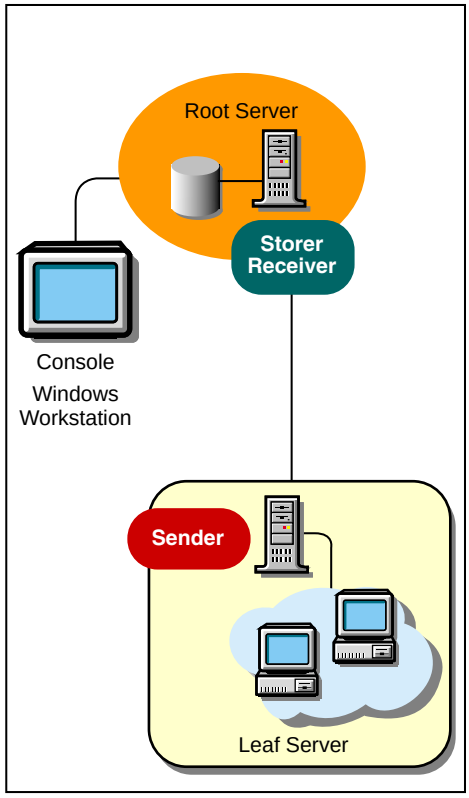
- Ensure that blank lines do not exist between the sections of the file.
- The file should end with a Carriage Return.
- Ensure that spaces and symbols in the *scanned_vendor_name_reported_by_the_scanner* and *scanned_product_name_reported_by_the_vendor* do not exist. The scanners compare the *scanned_vendor_name_reported_by_the_scanner* and the *scanned_product_name_reported_by_the_scanner* with the scanned data that they collect. Ensure that names that you use are not case-sensitive.

If you specify incorrect entries in the file, the entries preceding the incorrect entry will be used and the other entries will be ignored.

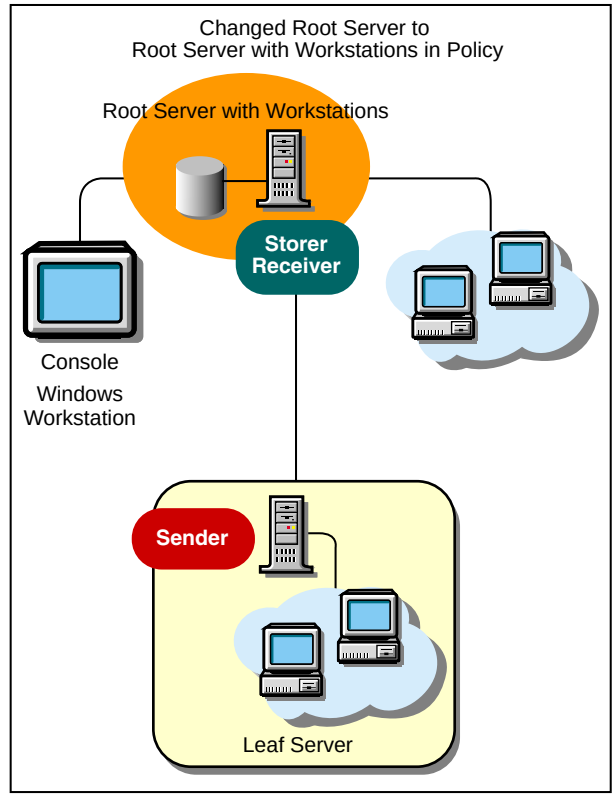
Changing the Role of the Inventory Service Object

When you install ZfD, you assign specific roles to the server based on your inventory deployment. For example, if the deployment plan identifies three servers, such as a Root Server, an Intermediate Server with Database, and a Leaf Server for inventory deployment, you install Workstation Inventory on these servers, and choose the role for the server. Later, if you want to make changes in the inventory deployment, such as attaching the workstations to the existing Root Server, you need to change the role of the Inventory Service object from Root Server to Root Server with Workstations. Additionally, depending on the new role, there are some policies you need to configure.

The following illustration shows the current deployment and the changes in the deployment:



Existing Inventory Deployment



Changes in the Inventory Deployment Through Policies

To change the role for any server:

1. Plan the change of roles carefully because the changes will impact the existing inventory deployment. Also, consider the disk space requirements and ensure that you have the required configurations for Inventory. For more information, see Workstation Inventory in *Deployment.*
2. In ConsoleOne, right-click the Inventory Service object (*servername*_ZenInvservice) > click Properties > click the Inventory Service Object Properties tab.
3. Choose the new role of the Inventory Service Object > click Apply.

   You will see a list of actions that you should follow based on the chosen role. For example, if you change the Root Server to a Root Server with Workstations, you need to configure the Workstation Inventory policy for the workstations that you have attached. Similarly, to change the role to

any other server, you need to follow the instructions to make the new role change effective.

4. Bring down the services running on the changed server, follow the actions that you need to change the role, and then bring up the server.

The following sections contain information to help you change the role of the Inventory Service object:

- "Changing the Role of the Root Server" on page 463
- "Changing the Role of the Root Server with Workstations" on page 465
- "Changing the Role of the Intermediate Server" on page 467
- "Changing the Role of the Intermediate Server with Database" on page 468
- "Changing the Role of the Intermediate Server with Database & Workstations" on page 469
- "Changing the Role of the Intermediate Server with Workstations" on page 470
- "Change the Role of the Leaf Server" on page 472
- "Changing the Role of the Leaf Server with Database" on page 473
- "Changing the Role of the Standalone Server" on page 474

Changing the Role of the Root Server

To change the role of the Root Server to a different role, follow the actions specified in the following table:

To change the role of the Root Server to ...	Tasks:
Root Server with Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached to the Root Server with Workstations will be scanned for. 2. Ensure that you check the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations attached to will be done.

To change the role of the Root Server to ...	Tasks:
Intermediate Server	Perform the following tasks: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server. 2. If a Database Location policy is associated with a Root Server, remove the policy.
Intermediate Server with Database	Perform the following task: 1. After changing the role, configure the Roll-Up to specify the next-destination server for roll-up of data from this server.
Intermediate Server with Database & Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for. 2. Ensure that you check the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations will be done. 3. Configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server.
Intermediate Server with Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for. 2. Ensure that you check the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations will be done. 3. Configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server. 4. If a Database Location policy is associated with the Root Server, remove the policy.

To change the role of the Root Server to ...	Tasks:
Leaf Server, Leaf Server with Database, or Standalone Server	Workstation Inventory does not allow you to change the Root Server to these servers because these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Workstation Inventory component.

Changing the Role of the Root Server with Workstations

Follow the actions specified in the following table:

To Change the Role of the Root Server with Workstations to ...	Tasks:
Root Server	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Root Server with Workstations or reconfigure the Inventory policy.
Intermediate Server	Perform the following tasks: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server. 2. If the Database Location policy is associated with the Root Server with Workstations, remove the policy.
Intermediate Server with Database	Perform the following tasks: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server. 2. Before changing the role, if the Workstation Inventory policy is associated with the Root Server with Workstations, remove the policy for those workstations attached to this server or to the lower-level servers that roll up to this server.

To Change the Role of the Root Server with Workstations to ...	Tasks:
Intermediate Server with Database & Workstations	Perform the following tasks: 1. Ensure that you check the Full Scan setting in the Inventory Service Object property page for the server. This setting will ensure that a complete scan of the workstations will be done. 2. Configure the Roll-Up policy to specify the next-destination server for roll-up of data from this server.
Intermediate Server with Workstations	Perform the following task: 1. Before changing the role, if the Database Location policy is associated with the Root Server with Workstations, remove the policy.
Leaf Server, Leaf Server with Database, or Standalone server	Workstation Inventory does not allow you to change the Root Server to these servers as these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Workstation Inventory component.

Changing the Role of the Intermediate Server

Follow the actions specified in the following table:

To Change the Role of the Intermediate Server to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. Configure the Database Location policy.

To Change the Role of the Intermediate Server to ...	Tasks:
Root Server with Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy for those workstations attached to this server. 2. Before changing to this role, remove the Roll-Up policy. 3. Configure the Database Location policy.
Intermediate Server with Database	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Database & Workstations	Perform the following tasks: 1. After changing the role, configure the Workstation Inventory policy so that all the workstations associated to this Inventory Service object, and also those workstations associated to the lower-level servers that roll up to this server will be scanned for. 2. Configure the Database Location policy.
Intermediate Server with Workstations	Perform the following task: 1. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for.
Leaf Server, Leaf Server with Database, or Standalone server	Workstation Inventory does not allow you to change the Intermediate Server to these servers as these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Workstation Inventory component.

Changing the Role of the Intermediate Server with Database

Follow the actions specified in the following table:

To Change the Role of the Intermediate Server with Database to ...	Tasks:
Root Server	Perform the following task: 1. Before changing to this role, remove the Roll-Up policy.
Root Server with Workstations	Perform the following tasks: 1. Before changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for. 2. Remove the Roll-up policy.
Intermediate Server	Perform the following task: 1. Before changing the role, if the Database Location policy is associated with the Intermediate Server with Database, remove the policy.
Intermediate Server with Database & Workstations	Perform the following task: 1. After changing the role, configure the Workstation Inventory policy so that the workstations attached will be scanned for.
Intermediate Server with Workstations	Perform the following tasks: 1. Before changing the role, if the Database Location policy is associated with the Intermediate Server with Database, remove the policy. 2. After changing the role, configure the Workstation Inventory policy so that the workstations that you have attached will be scanned for.
Leaf Server, Leaf Server with Database, or Standalone server	Workstation Inventory does not allow you to change the Intermediate Server to these servers as these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Workstation Inventory component.

Changing the Role of the Intermediate Server with Database & Workstations

Follow the actions specified in the following table:

To Change the Role of the Intermediate Server with Database & Workstations to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. Remove the Workstation Inventory policy associated with the server so that the workstations will not send the scan files to this server.
Root Server with Workstations	Perform the following task: 1. Before changing to this role, remove the Roll-Up policy.
Intermediate Server	Perform the following tasks: 1. Remove the Workstation Inventory policy associated with the lower-level servers that roll-up to the Intermediate Server with Database & Workstations server. 2. Remove the Database Location policy.
Intermediate Server with Database	Perform the following task: 1. Remove the Workstation Inventory policy of the Intermediate Server with Database & Workstations or reconfigure the policy.
Intermediate Server with Workstations	Perform the following task: 1. Before changing the role, remove the Database Location policy is associated with the Intermediate Server with Database & Workstations.

To Change the Role of the Intermediate Server with Database & Workstations to ...	Tasks:
Leaf Server, Leaf Server with Database, Standalone Server, or Intermediate Server with Workstations	Workstation Inventory does not allow you to change the Intermediate Server to these servers as these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Workstation Inventory component.

Changing the Role of the Intermediate Server with Workstations

Follow the actions specified in the following table:

To Change the Role of the Intermediate Server with Workstations to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. Remove the Workstation Inventory policy associated with the server so that the workstations attached will not send the scan files to this server. 3. After changing the role, configure the Database Location policy for the server.
Root Server with Workstations	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. After changing the role, configure the Workstation Inventory policy for those workstations attached to the lower-level server that roll up to this server. 3. Configure the Database Location policy

To Change the Role of the Intermediate Server with Workstations to ...	Tasks:
Intermediate Server	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy associated with the server.
Intermediate Server with Database	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy associated to the server attached to this Inventory Service object. 2. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Database & Workstations	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Leaf Server, Leaf Server with Database, Standalone Server, or Intermediate Server with Workstations	Workstation Inventory does not allow you to change the Intermediate Server to these servers as these changes affect the complete inventory system. If you want to assign these roles, you should reinstall and set up the Workstation Inventory component.

Change the Role of the Leaf Server

Follow the actions specified in the following table:

To Change the Role of the Leaf Server to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Roll-Up policy. 2. Remove the Workstation Inventory policy associated with the server. 3. After changing the role, configure the Database Location policy for the Root Server.
Root Server with Workstations	Perform the following tasks: 1. After changing the role, Configure the Database Location policy. 2. Remove the Roll-Up policy.
Intermediate Server	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy for those workstations associated with the server or reconfigure.
Intermediate Server with Database	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy for those workstations associated with the lower-level servers that roll up to this server or reconfigure the policy. 2. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Database & Workstations	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Intermediate Server with Workstations	This change of role does not require any specific policy modifications.

To Change the Role of the Leaf Server to ...	Tasks:
Leaf Server with Database	Perform the following task: 1. After changing the role, configure the Database Location policy for the server.
Standalone Server	Perform the following task: 1. Remove the Roll-up policy.

Changing the Role of the Leaf Server with Database

Follow the actions specified in the following table:

To Change the Role of the Leaf Server with Database to ...	Tasks:
Root Server	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Leaf Server with Database. 2. Remove the Roll-up policy.
Root Server with Workstations	Perform the following task: 1. Remove the Roll-up policy.
Intermediate Server	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy and the Database Location policy associated with the Leaf Server with Database.
Intermediate Server with Database	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Leaf Server with Database.
Intermediate Server with Database & Workstations	This change of role does not require any specific policy modifications.

To Change the Role of the Leaf Server with Database to ...	Tasks:
Intermediate Server with Workstations	Perform the following task: 1. Before changing to this role, remove the Database Location policy associated with the Leaf Server with Database.
Leaf Server	Perform the following task: 1. Before changing to this role, remove the Database Location policy associated with the Leaf Server with Database.
Standalone Server	Perform the following task: 1. Before changing to this role, remove the Roll-Up policy.

Changing the Role of the Standalone Server

Follow the actions specified in the following table:

To Change the Role of the Standalone Server to ...	Tasks:
Root Server	Perform the following task: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Standalone Server.
Root Server with Workstations	This change of role does not require any specific policy modifications.
Intermediate Server	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy and the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy.

To Change the Role of the Standalone Server to ...	Tasks:
Intermediate Server with Database	Perform the following tasks: 1. Before changing to this role, remove the Workstation Inventory policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from the Intermediate Server with Database.
Intermediate Server with Database & Workstations	Perform the following tasks: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from the Intermediate Server with Database & Workstations.
Intermediate Server with Workstations	Perform the following tasks: 1. Before changing to this role, remove the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from the Intermediate Server with Workstations.
Leaf Server	Perform the following tasks: 1. Before changing to this role, remove the Database Location policy associated with the Standalone Server. 2. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from the Leaf Server.

To Change the Role of the Standalone Server to ...	Tasks:
Leaf Server with Database	Perform the following task: 1. After changing the role, configure the Roll-Up policy to specify the next-destination server for roll-up of data from the Leaf Server with Database.

Assigning Trustees to the Database Object

You must have administrative privileges for any inventory operations from ConsoleOne. An administrator can assign the users as trustees of the database object.

To assign trustees to the database object:

1. Log in as an administrator to the tree and context on which you have installed ZfD3.
2. In ConsoleOne, right-click the existing database object > click Trustees of this object > click Add Trustee.
3. Browse to the user objects or container that contains the user objects > click the user objects that need to be the trustees or click the container > click OK.
4. Click All Attributes Rights > select Read rights > click OK twice.

Migrating ZENworks 2 Inventory Information

In ZfD3, you can use the existing inventory information of ZENworks 2.

If you have configured ZENworks 2 for a large number of workstations, you need not disrupt the inventory process when upgrading to ZfD3. You need to follow this procedure to continue using the inventory information:

1. Migrate the Inventory policies in ZfD3
2. Migrate the ZENworks 2 database

The following sections will help you migrate your ZENworks 2 inventory information to ZfD3:

- “Using ZENworks 2 Inventory Policies” on page 477

- “Migrating the Inventory Information from the ZENworks 2 Database” on page 478
- “Scenarios: Migration of ZENworks 2 Inventory Information” on page 482

After successfully migrating the policies and the database, the inventory information of ZENworks 2 should exist in ZfD3. Also, the inventory information in ZENworks 2 and ZfD3 can coexist. The workstation inventory is stored in the ZENworks 2 database as specified in the ZENworks 2 Inventory policies. The inventory information is also stored in the ZfD3 database. You can configure the Inventory policy settings to ensure a full scan of the workstations.

Using ZENworks 2 Inventory Policies

If you have a large number of workstations associated with Inventory policies in ZENworks 2, you can continue to use the same policies in ZfD3. Migrating ZENworks 2 Inventory policies is an easy method of importing the same policy settings for the associated workstations to ZfD3.

The Inventory policy migration tool associates the workstations that were associated with the Inventory policy in ZENworks 2. After you migrate, you may need to further configure the policy settings in ZfD3.

The Inventory policy migration tool does not remove the Inventory policies in ZENworks 2.

To migrate ZENworks 2 Inventory policies:

1. Migrate the Workstation Inventory Package in ZfD3. See Migrating ZENworks 2 Policies in Workstation Management in *Deployment*.
2. In ConsoleOne, click the Inventory Service object to which the workstations are attached.

   You must select an Inventory Service object that supports the role of an inventory server.
3. Click Tools > Inventory Policy Migration.
4. Specify the following options:

   Server Address IP/DNS: If your ZENworks 2 inventory server is a NetWare® 4.*x* server, specify the Server Address.

NDS Search Context: Specify the context for searching the Workstation Inventory object. By default, this tool will search the Workstation object in the current root context.

5 Click Find.

If any ZENworks 2 Inventory policies are found, these policies are listed in the Reports window.

6 Click Migrate.

All the listed Inventory policies will be migrated. You can see the list of the Inventory policies that were migrated in the Report window.

To ensure that the migration is successful, open the Inventory policy in ZfD.

In ConsoleOne, double-click the Inventory Service object for which you have migrated the policies > click the Workstation Inventory policy tab. You will see the same inventory settings as specified in ZENworks 2.

Migrating the Inventory Information from the ZENworks 2 Database

The Database Migration tool migrates the existing inventory information from the ZENworks 2 database to the ZfD3 database. The Migration tool provides an easy way to migrate a large amount of existing data into the ZfD3 database. Once you migrate the database, you can choose when you want to rescan the workstations in the inventory policies.

The migration tool migrates the following list of inventory information:

- Mouse, BIOS, bus, user, inventory scanner, keyboard, unitary computer system, monitor, software, processor, CD ROM, operating system, parallel port, serial port, general information, hard disk, floppy disk, NIC information, IP address and subnet mask, IPX information, DNS information, modem information, and memory information.

HINT: The migration tool does not migrate the DMI scan data.

When you install ZfD3, the migration tool that uses the DBMIGRATE.NCF file is installed on NetWare servers. This file is in the SYS:\SYSTEM directory on a NetWare server and in the PUBLIC\ZENWORKS\WMINV\BIN on a Windows NT/2000 server. On Windows NT/2000, the batch file, DBMIGRATE.BAT is installed.

The contents of the .NCF file are as follows:

```
envset
JDBC_DRIVER=SYS:\PUBLIC\ZENWORKS\LIB\JDBCDRV.ZIP;SYS:\PUBLIC
\ZENWORKS\LIB\CLASSES111.ZIP

envset
WORKING_PATH=SYS:\PUBLIC\ZENWORKS\WMINV\LIB\STATUSLOG.JAR;SY
S:\PUBLIC\ZENWORKS\WMINV\LIB\DESKTOPCOMMONUTILITY.JAR;
SYS:\PUBLIC\ZENWORKS\MWUTILITY.JAR

envset CLASSPATH=.;$JDBC_DRIVER;$WORKING_PATH;$CLASSPATH

#

# Running the DBMigrate Utility

# *********

# To run the DBMIGRATE Utility, do the following:

#

#1. Check if the environment variable JDBC_DRIVER points to
the path where JDBCDRV.ZIP is present.

#2. Check if the environment variable WORKING_PATH points to
the path where ZENINVSERVER.JAR, STATUSLOG.JAR, and
DESKTOPCOMMONUTILITY.JAR are present.

# 3. Enter the IP address ZENworks for Desktops 3 Inventory
database server after the switch -DBLOC. For example,
$CLASSPTH -dbloc 164.99.156.184

# 4. Enter the IP address of ZENworks 2 Inventory database
after the switch -zen2dbloc. For example, $classpath -dbloc
164.99.156.134 -zen2dbloc 164.99.156.135

# **********

# If the ZENworks for Desktops 3 Inventory database is running
on Oracle, then uncomment the below the line and comment the
Sybase line.

# java -ns
com.novell.zenworks.desktop.inventory.migration.database.Loa
der -classpath $classpath;$tmppath -nds -dbloc 164.99.135.198
-zen2dbloc 164.99.156.42 -oracle -sid abuorcl -Logfilename
sys:\etc\dbmigrte.log

# If the ZENworks for Desktops 3 Inventory database is running
on Sybase, then uncomment the below line and comment the above
Oracle line.
```

```
-classpath $classpath;$tmppath -nds -dbloc 164.99.156.42 -
zen2dbloc 164.99.145.53 -Logfilename SYS:\ETC\DBMIGRATE.LOG
```

You must modify the file to specify the location of the ZENworks 2 database server and the ZfD3 database and the path settings.

To migrate a database from ZENworks 2:

1. Ensure that the Inventory database is installed when you install ZfD3.
2. Stop the Inventory Gatherer in ZENworks 2.

   Also, ensure that all the .STR files are stored in the database. If the .STR file data is updated in the database, the directory containing the .STR files (ZENWORKS\STRDIR) will be empty. See the ZENworks 2 documentation Web site (www.novell.com/documentation) for more information about unloading the Inventory Gatherer.
3. In ZfD3, stop the Service Manager.
4. Modify the settings on the server as follows:
   - If the database sever is on a NetWare server, modify the settings in the DBMIGRATE.NCF file. At the database server console, enter **DBMIGRAT.NCF**.
   - If the database server is on a Windows NT/2000 server, at the database server console, run the DBMIGRAT batch file.

     dbmigrate *ipaddress_of_zfd3_database server ipaddress_of_zenworks2_database_server*

You will see a list of workstations that have been successfully migrated into the ZfD database.

The database migration tool creates a log file (DBMIGRATE.LOG) in the SYS:\ETC directory.

If you have installed more than one ZfD3 database, you must migrate the ZENworks 2 database to each database, repeating the database migration procedure for each database.

The database information from the ZENworks 2 database will not be removed.

Deleting the Inventory Information from ZENworks 2 database

You can remove the inventory information from the ZENworks 2 Inventory database, ZENINV.DB, using the Delete tool.

Before you run this tool, ensure that the ZENworks 2 and ZfD3 databases are not loaded. You can run this tool on any inventory server that has the ZENworks 2 and ZfD3 databases connected to the server.

To run this tool:

1. Modify the MGMTDBS.NCF file.

   1a. Insert the -d parameter in the MGMTDBS.NCF after the dvsrv7 switch.

   The contents of this file should be as follows:

   ```
   dbsrv7 -d -gm 50
   ```

   1b. Save the changes.

2. Load the ZfD3 database, at the server prompt, enter **mgmtdbs**.

   This will load the ZENworks 2 database (ZENINV.DB) also.

3. Stop the Naming server.

   On NetWare servers, at the inventory server prompt, view the ID of the Naming server, enter **JAVA -show**

   Enter **JAVA -KILL*id_of_the_naming_server***

4. At the server prompt, enter **zen2remove**.

   Displays that the inventory information is being deleted.

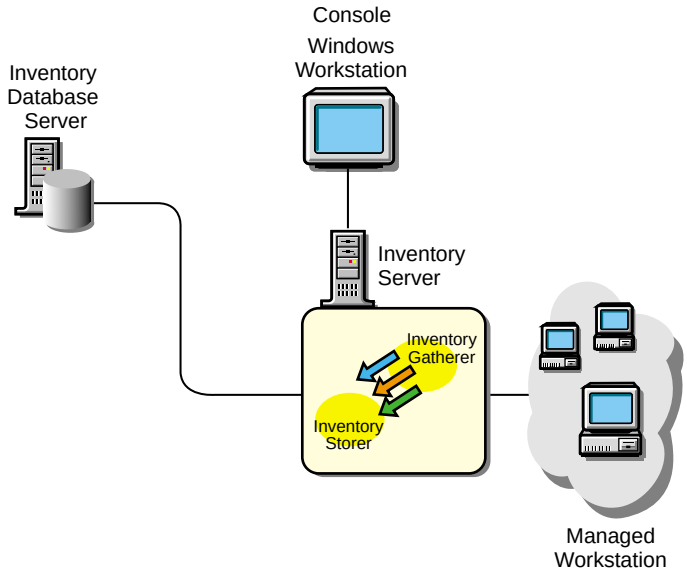
   On completing the removal of inventory information from the ZENINV.DB database, the status of deletion is displayed. To view the log file for the Delete tool, open the file DROPZEN2.LOG file in the SYS:\ETC directory.

5. Remove the modifications in the MGMTDBS.NCF file.

   Delete the -d parameter in the MGMTDBS.NCF file and save the changes.

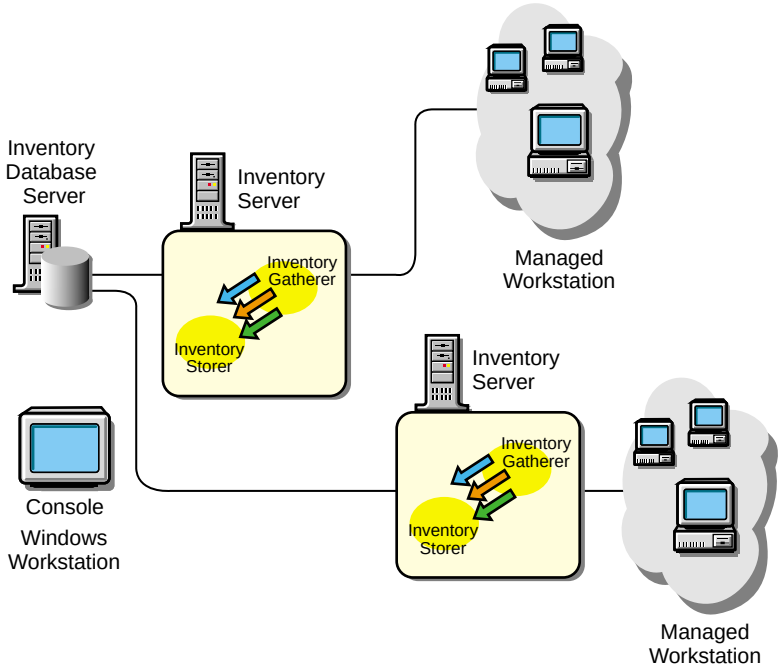
Scenarios: Migration of ZENworks 2 Inventory Information

In this type of inventory configuration in ZENworks 2, the inventory server components and the database are located on different servers. The scenario is illustrated in the following figure:



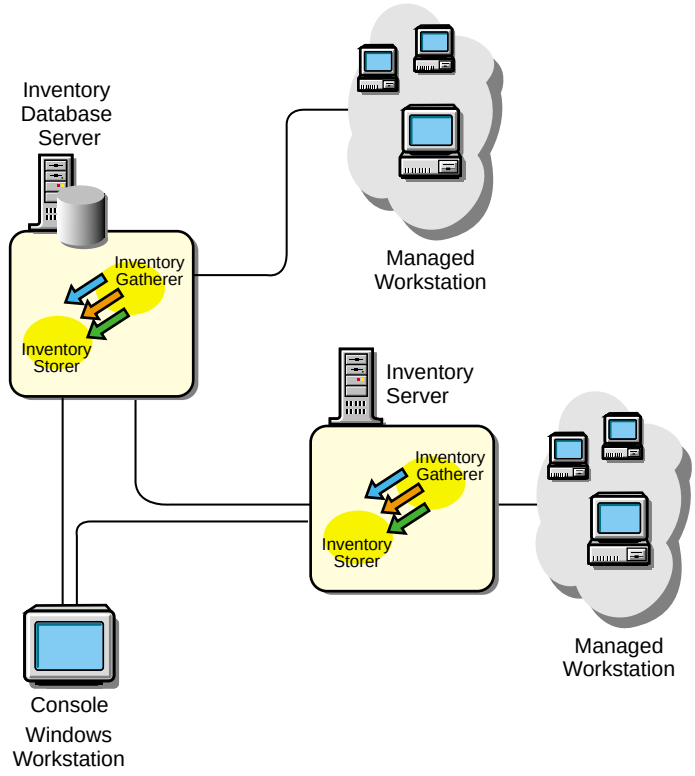
To migrate the inventory information, you need to migrate the Workstation Package and the Workstation Inventory policies associated with the inventory server. Later you migrate the Inventory database and configure the Database Location policy.

In another type of inventory configuration in ZENworks 2, there is more than one inventory server. These inventory servers are connected to one Inventory database server. The scenario is illustrated in the following figure:



To migrate the inventory information, you migrate the Workstation Package and the Workstation inventory policies associated with each inventory server. Later you migrate the Inventory database and configure the Database Location policy.

In the following inventory configuration in ZENworks 2, there is more than one inventory server. Also, an inventory server has an Inventory database; the other inventory servers connect to this inventory server with the database. The scenario is illustrated in the following figure:



To migrate the inventory information, you need to migrate the Workstation Package and the inventory policies associated with the inventory server with a database. Later, you migrate the other inventory servers. You must stop the Inventory components (Inventory Gather and Inventory Storer) on the server.

Setting up the Inventory Database

The following sections contain detailed information to help you set up your Inventory database:

- “Loading the Inventory Database as a Separate Oracle Instance” on page 485
- “Optimizing the Performance of the Oracle Database” on page 498
- “Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool)” on page 499

- “Using an Optimal Database Cache Size on the Inventory Database Server to Improve Performance” on page 501
- “Using the Inventory Database Service Configuration Tool for Windows NT/2000” on page 502
- “Deleting the Inventory Information from the Inventory Database” on page 503
- “Synchronizing the Inventory Database with NDS (NDS-DB Sync Tool)” on page 503
- “Backing Up the Inventory Database” on page 504

Loading the Inventory Database as a Separate Oracle Instance

The following sections explain the steps for configuring and running multiple Oracle 8i database instances:

- “Configuring and Running Multiple Oracle Database Instances on a NetWare 5.x Server” on page 485
- “Configuring and Running Multiple Oracle Database Instances on a Windows NT/2000 Server” on page 492

Configuring and Running Multiple Oracle Database Instances on a NetWare 5.*x* Server

Prerequisites for configuring the database include the following:

- Oracle 8i (8.1.5.0.4) or later Enterprise Edition on a NetWare 5.*x* server must be installed on the server before configuring the Inventory database.
- Workstation Inventory requires that you have a minimum of five user licenses.
- You have already installed ZfD3 Workstation Inventory. For more information, see Workstation Inventory in *Getting Started*.

Follow these instructions:

1. Unload Oracle, at the database server prompt, enter **`oraunld`**.
2. Invoke the Net8 configuration utility. At the database server prompt, run **`easycfg.ncf`**.

   This loads the Net8 Easy configuration window.
3. Define a unique Oracle instance.

   3a Click Config > Listener > Database.

3b Click Add.

3c Assign values for Database Instance and Database Name in the Adding Instances Address window.

For example, assign Database Instance=*zfd3 and Database Name=mgmtdb*.

In this configuration, the database instance is zfd3. You can specify any database instance name.

The Database Domain field should be left blank.

3d Click Accept > Save.

4 Configure the Listener for IPC.

To run an Oracle system, the IPC and TCP addresses should be already be configured.

4a Click Config.

4b Click Listener >Address.

Ensure that IPC and TCP addresses are configured for the server.

The settings should be as follows:

IPC: *servername*_LSNR

TCP: *IPaddress or hostname*

If these settings exist, click Cancel. Otherwise, assign the values for these settings > click Save.

5 Create an IPC alias.

5a Click Config > Database Alias.

The window will list the aliases for IPC, SPX, TCP, and others.

5b Click Add to add an alias name for the new instance.

5c Enter the following details:

Database Alias: *servername-databaseinstance-IPC*

For example, the database alias is *austr*, where *austr* is the servername, *zfd3* is the database instance created earlier.

Protocol: *IPC*

Service/Host Name or Key Name: *server_name_LSNR*

Database Instance: zfd3

5d Click Accept > Save.

To verify the configured alias name in the list window:

- Click Config > Database Alias > select the newly created alias.
- Click View.
- View the properties of the database alias. Ensure that the properties are correct.

  If the property settings are incorrect, delete the alias (click Delete) and repeat the from Step 5a on page 486 to Step 5d on page 486.

6 Exit the EasyCfg tool. Click Config > Exit.

7 Create a password file for logging as *Internal* user for this instance, enter **load orapwd81 file=*oracle_volume:oracle_home\DATABASE\PWDdatabase_instance.ORA password=password entries=2***

where *oracle_volume* is the NetWare volume name of your Oracle installation, *PWDdatabase_instance*.ORA is the password file name, and *password* is any password that you specify.

For example, this command is as follows: **load orapwd81 FILE=oracle:\orahome1\database\pwdzfd3.ora** *password=mgmtdb* entries=*2*

This password file will be created in the *oracle_volume:*\DATABASE directory. Ensure that the file exists in the directory.

8 Load the Oracle NLMs. At the database server prompt, enter **oraload**.

9 Set the newly created ZfD instance.

9a Load the Oracle Server Manager. At the database server prompt, enter **svrmgr31**.

Enter the following commands:

set instance *servername-databaseinstance-IPC*.

For example, **set instance *austr-zfd3-ipc***

This displays that the newly created ZfD3 instance is started.

connect internal/*password*

where, *password* is the password created in Step 7 on page 487.

9b Mount the Inventory database.

- Extract the INVORACLE8INW.ZIP file from the \ZENWORKS\PRODUCTS\DATABASE\ORACLE directory in the *ZENworks for Desktops* product CD to any volume other than SYS: on the NetWare server.

  After extraction, you will see the following list of system files and Database Space files in this directory.

Filenames	Contents Description
CIM1.ORA, CIM2.ORA, CIM3.ORA, CIM4.ORA, CIM5.ORA, CIM6.ORA, CIM7.ORA, CIM8.ORA, and CIM9.ORA	Table spaces that contain the inventory information for ZfD
CTL1.ORA	Oracle control file
INIT.ORA	Database startup file
LOG1.ORA, LOG2.ORA	Online Redo log file
RBS1.ORA	Rollback segment file containing table space for rollback
SYS1.ORA	SYS schema file containing the data dictionary
TMP1.ORA	Temporary table space
_DBINT.SQL	Contains the Inventory database site name
_START.SQL	Contains the location of the database startup file
ALTERCTRL.SQL	Contains the location of table spaces
MGMTDBO.NCF	ZENworks Inventory database startup file for Oracle on NetWare
\TRACE*oracle_log_files*	Contains the Oracle log files

- Copy the MGMTDBO.NCF file from *volume*\ZFD3\ORACLE to the SYS:\SYSTEM directory on the database server, where *volume* is the volume on which the .ZIP file was extracted.
- Specify the site details of the database.

  Copy the _DBINIT.SQL file from *volume*\ZFD3\ORACLE to the SYS:\SYSTEM directory on the database server, where *volume* is the volume on which the .ZIP file was extracted.
- Edit the following lines in the _DBINIT.SQL file to specify the site ID and site name for the database:

  siteid:=*site_ID;*

  siteName:=*'site_name';*

  where, *site_ID* uniquely identifies the database and '*site_name*' provides a description for the database site.

  If you do not specify the site ID, the default values specified in the file apply.

  Save the changes you make.
- Create the Oracle Control file.

  Edit the ALTERCTRL.SQL file located in *volume*\ZFD3\ORACLE to specify the path to which you have extracted the .ZIP file, where *volume* is the volume on which the .ZIP file was extracted.

  For example, modify the existing DATA:\ZFD3\ORACLE\DATABASE path to ORACLE:\ZFD3\ORACLE\DATABASE in ALTERCTRL.SQL.

  In this .SQL file, modify the path for these parameters, if required.

  startup nomount pfile=*oracle:\zfd3\oracle\database\INIT.ORA*

  logfile group 1 *'oracle:\zfd3\oracle\database\log1.ora' size 256K,*

  logfile group 2 *'oracle:\zfd3\oracle\database\log2.ora' size 256K*

  datafile *'oracle:\zfd3\oracle\database\sys1.ora',*

  'oracle:\zfd3\oracle\database\rbs1.ora',

'oracle:\zfd3\oracle\database\cim1.ora',

'oracle:\zfd3\oracle\database\cim2.ora',

'oracle:\zfd3\oracle\database\cim3.ora',

'oracle:\zfd3\oracle\database\cim4.ora',

'oracle:\zfd3\oracle\database\cim5.ora',

'oracle:\zfd3\oracle\database\cim6.ora',

'oracle:\zfd3\oracle\database\cim7.ora',

'oracle:\zfd3\oracle\database\cim8.ora',

'oracle:\zfd3\oracle\database\cim9.ora',

'oracle:\zfd3\oracle\database\tmp1.ora'

Save the changes.

- Edit the INIT.ORA file located in *volume*\ZFD3\ORACLE\DATABASE to specify the path to which you have extracted the .ZIP file.

  where *volume* is the volume on which the .ZIP file was extracted.

  Modify the path for the following parameters:

  control_files=*location_of_CTLl.ORA\CTL1.ORA*

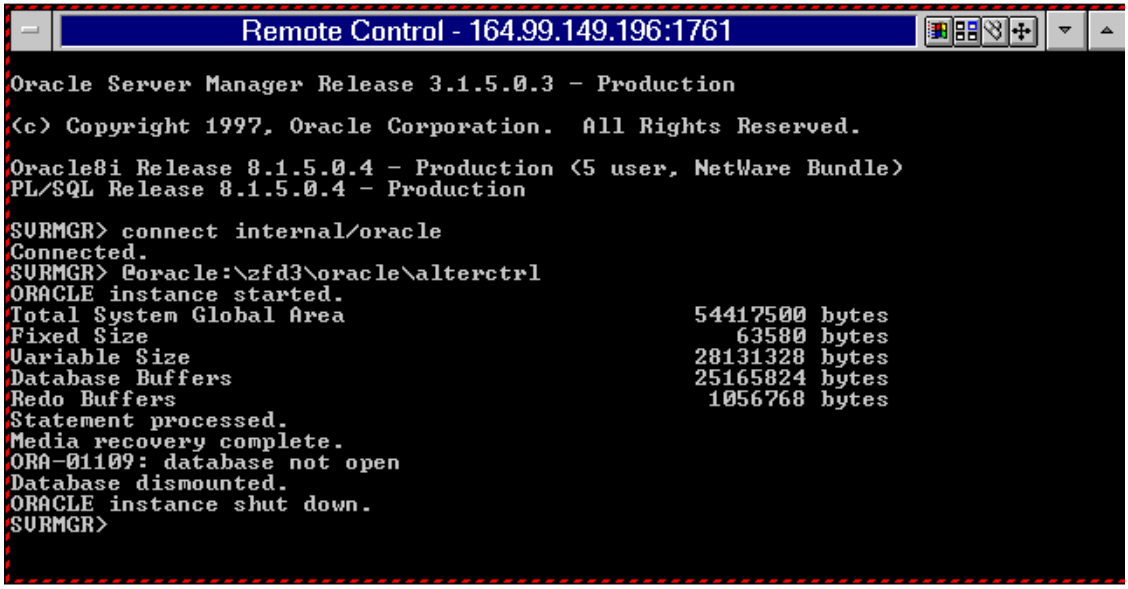
  background_dump_dest=*location_of_TRACE_dir\TRACE*

  user_dump_dest=*location_of_TRACE_dir\TRACE*

  Save the changes.

  At the Server Manager prompt, enter ***@complete_path_of_alterctrl.sql*\ALTERCTRL**.

You should see the following window, which indicates that the database is mounted and loaded.


Remote Control - 164.99.149.196:1761
Oracle Server Manager Release 3.1.5.0.3 - Production
(c) Copyright 1997, Oracle Corporation. All Rights Reserved.
Oracle8i Release 8.1.5.0.4 - Production (5 user, NetWare Bundle)
PL/SQL Release 8.1.5.0.4 - Production
SVRMGR> connect internal/oracle
Connected.
SVRMGR> @oracle:\zfd3\oracle\alterctrl
ORACLE instance started.
Total System Global Area 54417500 bytes
Fixed Size 63580 bytes
Variable Size 28131328 bytes
Database Buffers 25165824 bytes
Redo Buffers 1056768 bytes
Statement processed.
Media recovery complete.
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
SVRMGR>

Exit the Server Manager, enter **exit**.

10 Modify the _START.SQL file located in SYS:\SYSTEM.

10a Enter the following line in the file:

```
set instance servername-databaseinstance-IPC

shutdown normal
```

11 Modify the properties of the Database object.

11a Create the Database object. In ConsoleOne, right-click a location in the tree for the Database object > click New > click Object > click ZENworks Database > click OK. Type a name for the Database object > click OK.

11b Configure the Database Server options of the Database object. In ConsoleOne, right-click the Database object > click Properties > click ZENworks Database. Browse for the DN of the server or type the IP address of the server. Type the values for the following options: Database(Read-Write) User Name: *MW_DBA* and Database(Read-Write) Password: *novell*. Click OK.

11c Ensure that the JDBC Driver properties are correct as per your database configuration. In ConsoleOne, right-click the Database object > click Properties > click Jdbc Driver Information. Click

Populate Fields with Default Values for an Oracle Database > click Populate Now. The settings for the Oracle database should be as follows:

Driver: *oracle.jdbc.driver.OracleDriver*

Protocol: *jdbc:*

SubProtocol: *oracle:*

SubName: *thin:@*

Port:*1521*

SID Service Name (Service ID of the Oracle database): *zfd3*

The value for the SID is the same as assigned for the Database Instance, see Step 5 on page 486.

Click OK.

12 At the database server prompt, enter **`mgmtdbo`**.

Enter the password created in Step 7 on page 487.

13 Start other Oracle database instances. Run the Oracle Server Manager, set the applications instance, and mount the database.

HINT: If you are loading multiple databases in separate Oracle instances, then each database reserves a separate Oracle SGA memory, where Oracle keeps all the database resources. In such environments, you should increase the amount of memory on the server. Refer to the documentation provided by Oracle.

Configuring and Running Multiple Oracle Database Instances on a Windows NT/2000 Server

Prerequisites for configuring the database include the following:

- Oracle 8i (8.1.5.0.4) Enterprise Edition must be installed on the Windows NT/2000 server before configuring the Inventory database.
- To maintain the Inventory database in Oracle, Workstation Inventory requires that you have a minimum of five user licenses.
- You have already installed ZfD3 Workstation Inventory. For more information, see Workstation Inventory in *Getting Started*.

Follow these instructions:

1. At the database server, run the Oracle Database Configuration Assistant. From the desktop Start menu, click Programs > Oracle > Database Administration > Oracle Database Configuration Assistant.
2. Click Create a Database > Next.
3. Click Typical > Next.
4. Click Copy existing database files from the CD > click Next.
5. Enter the following details:

   Global Database Alias: **`mgmtdb.`*`your_windows_nt/2000_name`***.

   SID: The value is automatically filled as *mgmtdb*.
6. Click Finish.

   This allows for Oracle database creation. This process takes a significant amount of time. Ensure that the OracleServiceMGMTDB service is created and started.
7. Load the Inventory database.

   7a Run the Oracle Server Manager. From the desktop menu, click Start > Run > SVRMGRL.

   7b Enter the following commands:

   ```
   set instance mgmtdb
   connect internal/password_for_administrator
   ```

   7c Mount the Inventory database.

   - Extract the INVORACLE8INT.ZIP file from the \ZENWORKS\PRODUCTS\DATABASE\ORACLE directory located on the *ZENworks for Desktops* product CD to a drive, which preferably has NTFS.

     After extraction, you will see the following list of system files and Database Space files in this directory:

Filenames/Directory	Contents Description
CIM1.ORA, CIM2.ORA, CIM3.ORA, CIM4.ORA, CIM5.ORA, CIM6.ORA, CIM7.ORA, CIM8.ORA, and CIM9.ORA	Table spaces that contain the inventory information for ZfD

Filenames/Directory	Contents Description
CTL1.ORA	Oracle control file
INIT.ORA	Database startup file
LOG1.ORA, LOG2.ORA	Online Redo log file
RBS1.ORA	Rollback segment file containing table space for rollback
SYS1.ORA	SYS schema file containing the data dictionary
TMP1.ORA	Temporary table space
_DBINT.SQL	Contains the Inventory database site name
_START.SQL	Contains the location of the database startup file
ALTERCTRL.SQL	Contains the location of table spaces
MGMTDBO.BAT	ZENworks Inventory database startup file for Oracle on Windows NT/2000
\TRACE*oracle_log_files*	Contains the Oracle log files

- Specify the site details of the database.
- Edit the following lines in the _DBINIT.SQL file to specify the site ID and site name for the database.

  siteid:=*site_ID;*

  siteName:=*'site_name';*

  where, *site_ID* uniquely identifies the database, and *'site_name'* provides a description for the database site.

  If you do not specify a site ID, the default values specified in the file apply.

- Save the changes you make.

- Create the Oracle Control file.
- Edit the ALTERCTRL.SQL file located in *drive*\ZFD3\ORACLE to specify the path strings to which you have extracted the file, where *drive* is the drive on which the .ZIP file was extracted.

  For example, modify the existing *D:\ZFD3\ORACLE\DATABASE* path to *C:\ZFD3\ORACLE\DATABASE* in ALTERCTRL.SQL.
- In this .SQL file, modify the path for these parameters, if required.

  startup nomount pfile=*c:\zfd3\oracle\database\INIT.ORA*

  logfile group 1 *'c:\zfd3\oracle\database\log1.ora' size 256K*,

  logfile group 2 *'c:\zfd3\oracle\database\log2.ora' size 256K*

  datafile

  'c:\zfd3\oracle\database\sys1.ora',

  'c:\zfd3\oracle\database\rbs1.ora',

  'c:\zfd3\oracle\database\cim1.ora',

  'c:\zfd3\oracle\database\cim2.ora',

  'c:\zfd3\oracle\database\cim3.ora',

  'c:\zfd3\oracle\database\cim4.ora',

  'c:\zfd3\oracle\database\cim5.ora',

  'c:\zfd3\oracle\database\cim6.ora',

  'c:\zfd3\oracle\database\cim7.ora',

  'c:\zfd3\oracle\database\cim8.ora',

  'c:\zfd3\oracle\database\cim9.ora',

  'c:\zfd3\oracle\database\tmp1.ora',
- Save the changes.
- Edit the INIT.ORA located in *drive*\ZFD3\ORACLE\DATABASE, where *drive* is the drive on which the .ZIP file was extracted.

  Modify the path for the following parameters:

control_files=*location_of_CTLl.ORA\CTL1.ORA*

background_dump_dest=*location_of_TRACE_dir\TRACE*

user_dump_dest=*location_of_TRACE_dir\TRACE*

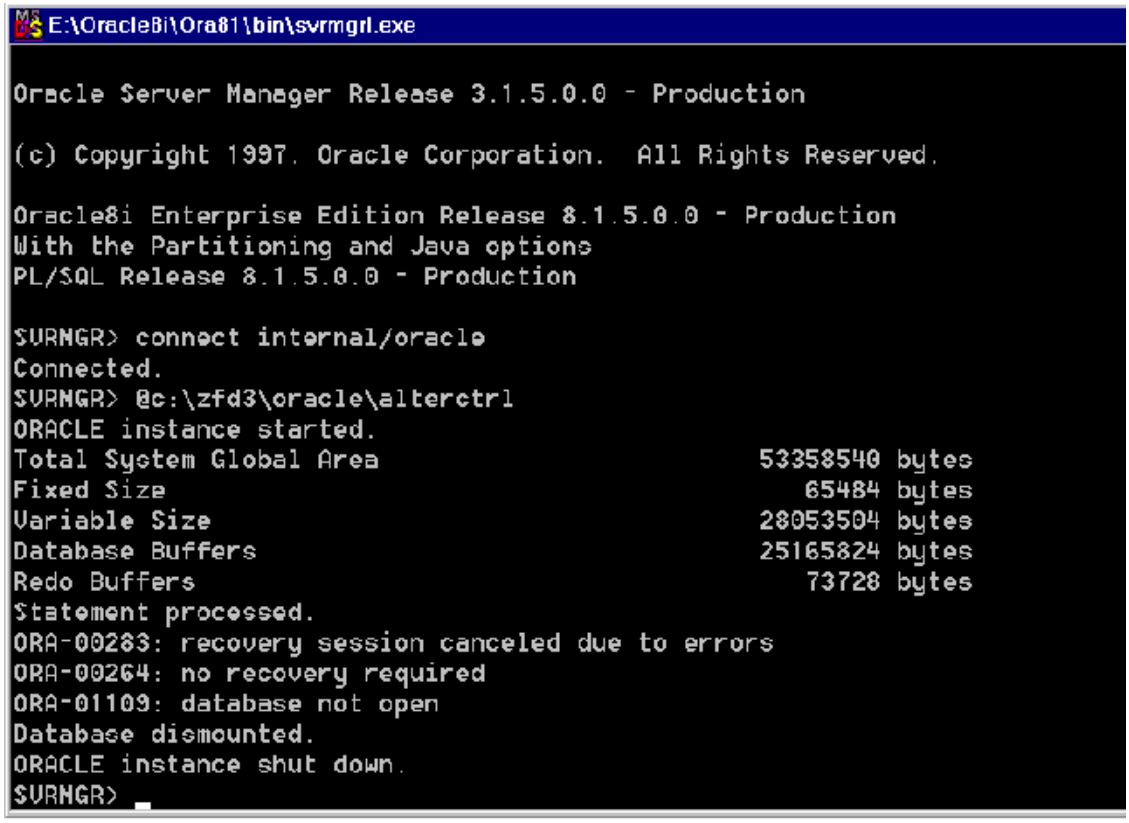
- Save the changes.
- Ensure that Oracle is loaded on the database server.
- Load the Oracle Server Manager. Click Start from the Windows taskbar > click Run > enter **`svrmgrl`**.

  Connect as an administrator. For example, if the administrator's internal name is *internal*, at the Server Manager prompt, enter **`connect internal/`*`password_for_administrator`***.

  Enter **`shutdown normal`**.

  At the Server Manager prompt, enter ***`@complete_path_of_alterctrl.sql`*`\ALTERCTRL`**.

  You should see the following window, which indicates that the database is mounted and loaded.


E:\Oracle8i\Ora81\bin\svrmgrl.exe
Oracle Server Manager Release 3.1.5.0.0 - Production
(c) Copyright 1997, Oracle Corporation. All Rights Reserved.
Oracle8i Enterprise Edition Release 8.1.5.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.5.0.0 - Production
SVRMGR> connect internal/oracle
Connected.
SVRMGR> @c:\zfd3\oracle\alterctrl
ORACLE instance started.
Total System Global Area 53358540 bytes
Fixed Size 65484 bytes
Variable Size 28053504 bytes
Database Buffers 25165824 bytes
Redo Buffers 73728 bytes
Statement processed.
ORA-00283: recovery session canceled due to errors
ORA-00264: no recovery required
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
SVRMGR>

Exit the Server Manager, enter **exit**.

8 Copy *drive*\ZFD3\ORACLE_START.SQL to the \ZFD3\ORACLE on the database server, where *drive* is the drive on which the .ZIP file was extracted.

8a Modify the _START.SQL file. Enter the following line in the file:

```
set instance mgmtdb

shutdown normal
```

9 Modify the properties of the database object.

9a Create the Database object. In ConsoleOne, right-click a location in the tree for the Database object > click New > click Object > click ZENworks Database > click OK. Type a name for the Database object > click OK.

9b Configure the Database Server options of the Database object. In ConsoleOne, right-click the Database object > click Properties > click ZENworks Database. Browse for the DN of the server or type the IP address of the server. Type the values for the following options: Database(Read-Write) User Name: *MW_DBA* and Database(Read-Write) Password: *novell*. Click OK.

9c Ensure that the JDBC Driver properties are correct as per your database configuration. In ConsoleOne, right-click the Database object > click Properties > click Jdbc Driver Information. Click Populate Fields with Default Values for an Oracle Database > click Populate Now. The settings for the Oracle database should be as follows:

Driver: *oracle.jdbc.driver.OracleDriver*

Protocol: *jdbc:*

SubProtocol: *oracle:*

SubName: *thin:@*

Port:*1521*

SID Service Name (Service ID of the Oracle database): *mgmtdb*

The value for the SID is the same as assigned for the Database Instance, see Step 5 on page 493.

Click OK.

10 At the database server console, from the path where the MGMTDBO.BAT(\ZFD3\ORACLE) is located, enter **mgmtdbo**.

Enter the password for the Database Administrator (DBA).

11 Start other Oracle database instances. Run the Oracle Server Manager, set the applications instance, and mount the database.

HINT: If you are loading multiple databases in separate Oracle instances, then each database reserves a separate Oracle SGA memory, where Oracle keeps all the database resources. In such environments, you should increase the amount of memory on the server. Refer to the documentation provided by Oracle.

Optimizing the Performance of the Oracle Database

If you have an Inventory database on Oracle, you can improve the performance of the database when you generate the inventory reports or query the database.

You use the database buffer cache to store the most recently used data blocks. The database cache is determined as DB_BLOCK_BUFFERS * DB_BLOCK_SIZE. These parameters are specified in the INIT.ORA file in the ZENWORKS\DATABASE directory on the database server.

DB_BLOCK_BUFFERS specifies the number of database buffers. DB_BLOCK_SIZE specifies the size of each database buffer in bytes.

The size of each buffer in the buffer cache is equal to the size of the data block.

Oracle recommends that the database buffer cache for any Online Transaction Processing Application (OLTP) should have a hit ratio of about 90%, which is optimal.

The ZfD Inventory database on Oracle has an approximate 88% hit ratio with a database cache size of 24 MB for 128 MB RAM, which is about 20% of total memory.

If there is additional memory, you configure the database cache size by increasing the DB_BLOCK_BUFFERS parameter in the INIT.ORA file.

Organizing the Database Spaces for a Sybase Database on NetWare or Windows NT/2000 Servers (AlterDBSpace Tool)

If there are more volumes on the multiple physical disks of the database server, placing the Sybase database space files on separate volumes improves the performance while accessing the database.

When you install the Sybase database component of ZfD3, the System Database file and the Database Spaces are installed in the volume on the database server you specify. On loading the Inventory database server, the System Database file (MGMTDB.DB) is loaded. This MGMTDB.DB file references the inventory information in the Database Spaces files. The Database Spaces files (MGMTDB1.DB, MGMTDB2.DB, MGMTDB3.DB, MGMTDB4.DB, MGMTDB5.DB, MGMTDB6.DB, MGMTDB7.DB, MGMTDB8.DB and MGMTDB9.DB) contain the inventory information.

The ALTERDB.PROPS file is installed on the database server in the PUBLIC\ZENWORKS\WMINV\PROPERTIES. You modify the sections in the file to specify the location of the Database Spaces on the volumes.

The contents of the ALTERDB.PROPS is as follows:

```
#Database Space Properties

count=9

mgmtdb1=volume_location_of_mgmtdb1

mgmtdb2=volume_location_of_mgmtdb2

mgmtdb3=volume_location_of_mgmtdb3

mgmtdb4=volume_location_of_mgmtdb4

mgmtdb5=volume_location_of_mgmtdb5

mgmtdb6=volume_location_of_mgmtdb6

mgmtdb7=volume_location_of_mgmtdb7

mgmtdb1=volume_location_of_mgmtdb8

mgmtdb1=volume_location_of_mgmtdb9

.....
```

To organize the database spaces:

1. Ensure that the database is not loaded.

2 Ensure that the Inventory Service Manager is not running on the server.

3 Manually move the Database Space files on the server volume.

Move the Database Space files in the following way for better performance:

- MGMTDB1 and MGMTDB2 on the same volume
- MGMTDB3 and MGMTDB6 on the same volume
- MGMTDB5 and MGMTDB7 on the same volume
- MGMTDB8 and MGMTDB4 on the same volume
- MGMTDB9 on a volume

4 Modify the location of the nine Database Space files in the ALTERDB.PROPS file.

For example, enter **mgmtdb3=*SYS:**ZENWORKS**DATABASE***

5 Load the database. Enter **mgmtdbs** on NetWare servers. On Windows NT/2000 servers, run the database service.

5a Stop the Naming server.

On NetWare servers, at the inventory server prompt, view the ID of the Naming server, enter **JAVA -show**.

Enter **JAVA -KILL*id_of_the_naming_server***.

On Windows NT/2000, close the Naming server window.

Ignore the error messages displayed on the console. These messages are displayed because the database space files are not loaded.

6 Ensure that the Database Location policy has been configured.

7 On the server console, run the AlterDBSpace service. Enter **AlterDBSpace**.

On the inventory server, the AlterDBSpace tool runs as a service.

You will see a message that the database is adjusted.

8 Exit the database and then load the database.

Ensure that while loading the database there are no errors. Errors indicate that the specified volume location of the database space files are incorrect or do not exist. Ensure that the path of the Database Spaces are correct in the ALTERDB.PROPS file and repeat the procedure to organize the database spaces.

IMPORTANT: If you place the Database Spaces in different volumes, the log file should be placed in the same volume as the System database file (MGMTDB.DB).

Using an Optimal Database Cache Size on the Inventory Database Server to Improve Performance

Improving the database cache size improves database performance.

You can improve the performance of the Inventory database maintained in Sybase on NetWare or Windows NT/2000 servers. The default database cache size is 16 MB; however, this database cache size may not be adequate for large databases with more than 10,000 workstations.

You should change the database cache size to an optimum size. We recommend a database cache size that is one-fourth of the database size. You must also consider server memory size while assigning a cache size. For example, if you have 128 MB RAM, then a cache size of 32 MB is recommended.

To change the database cache size on a NetWare server:

1. Close all connections to the Inventory database.
2. Quit the Sybase server.
3. Open the MGMTDB.NCF file in the SYS:\SYSTEM directory.
4. Modify the -c parameter.

   For example, -c 64M sets the cache size to 64 MB.
5. Save the file.
6. On the server console, load the Inventory database. Enter **`MGMTDBS.`**

To change the database cache size on a Windows NT/2000 server:

1. Run the NTDBCONFIGURE.EXE file from PUBLIC\ZENWORKS.
2. Modify the -c parameter.
3. Save the file.
4. Restart the server so that the Inventory database service (Adaptive Service Anywhere - ZENworks for Desktops 3) starts up.

Using the Inventory Database Service Configuration Tool for Windows NT/2000

The Inventory Database service configuration tool lets you modify the startup parameters to optimize the Sybase database on Windows NT/2000 servers.

To run this tool:

1. Run NTDBCONFIG.EXE located in the ZENWORKS directory on the database server.
2. Modify the parameters as per the Sybase specification.

   Ensure that you specify valid values for the parameters.
3. Click OK.
4. Restart the database service (ASA Adaptive Anywhere).

Sybase Database Startup Parameters

The parameters are as follows:

- **-c:** Sets the initial memory reserves for caching database pages and other server information. For example, -c 32M reserves 32 MB cache size.
- **-gc:** Sets the maximum length of time in minutes that the database server runs without doing a checkpoint on each database. The default value is 60 minutes. For example, -gc sets the checkpoint time as 120 minutes.
- **-m:** Deletes the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server.
- **-n:** Specifies the name of the database server. For example, -n WARTHOG names the server WARTHOG.
- **-ti:** Disconnects the connections that have not submitted a request for a certain number of minutes. The default is 240 (4 hours). A client machine in the middle of the database transaction locks until the transaction ends or the connection terminates. The -ti option is provided to disconnect inactive connections and to free their locks. For example, specify -ti 400.
- **-x:** Specifies a communication link. For example, -x tcpip indicates a TCP/IP link.

Deleting the Inventory Information from the Inventory Database

If you delete the Workstation object in ConsoleOne, the inventory information for that Workstation object is deleted from the Inventory database server it is attached to. Also, the inventory information is deleted from all the next-level servers for that particular Inventory database server.

Synchronizing the Inventory Database with NDS (NDS-DB Sync Tool)

ZfD provides a database sync tool to synchronize the Workstation objects stored in the Inventory database with the Workstation objects in NDS. This tool removes the workstations that do not exist in NDS from the Inventory database. The excess workstations in the Inventory database exist because these workstations may have been deleted from NDS; however, the corresponding workstations were not removed from the database.

Use this tool regularly to maintain the database in a consistent state with NDS. You must run this tool for each Inventory database.

HINT: You must ensure that the Service Manager is loaded when you run the Inventory database sync tool.

To run this tool:

1. Generate a lookup file in the NDS Lookup Phase.

   ZfD compares the list of workstations in the database with those in NDS and generates a lookup file. This file contains the list of workstations to remove from the database.

   The server property file contains the [NDSLookupForDB Service] section with the lookup filename in the ARGUMENTS parameter. The default lookup filename (WSDELETE.LOK) file is located in the PUBLIC\ZENWORKS\WMINV\LOG directory.

   The lookup filename in this section and the [DBDelete Service] section of the server property file should be the same.

   1a. At the inventory server console prompt, enter:

   ```
   StartSer NDSLookupForDB
   ```

2. Delete workstations in the lookup file from the Inventory database in the Database Delete Phase.

   2a. In the [DBDelete Service] section of the properties file, specify the lookup filename.

2b At the inventory server console, enter:

```
StartSer DBDelete
```

The WSDELETE.LOG log file located in the \PUBLIC\ZENWORKS\WMINV\LOGS directory contains the status of deletion. This file contains information as to whether the database was synchronized successfully with NDS.

Backing Up the Inventory Database

ZfD provides an option to back up the Inventory database from the server. We recommend that you back up the database on a weekly basis. However, if you are tracking the inventory of workstations frequently, increase the frequency of backup.

The following sections contain additional information to help you back up the Inventory database.

- "Backing Up the Inventory Database (Sybase)" on page 504
- "Backing Up the Inventory Database (Oracle)" on page 505

Backing Up the Inventory Database (Sybase)

You can back up the database files and the transaction log to the location relative to the SCANDIR path.

To use this tool:

- Run the backup tool on the inventory server to which you have associated a database server. If you deployed more than one database server, you must run the backup tool for each database server.
- The database that you back up should be configured in the Database Location policy.
- In the [DBBackup Service] section of the server property file, you modify the location of the backup destination in the ARGUMENTS parameter. The backup files are relative to the SCANDIR path. For example, if the SCANDIR path is SYS:\ZENWORKS\SCANDIR, the database is backed up in SYS:\ZENWORKS\DATABASE*directory_you_specify*. You must the modify the server property file located on the server on which you are running the backup tool. Modify the server property file corresponding to the role of the server. For example, if you are running

the backup tool on the Leaf Server with Database, modify the server property file, LEAF_DB_WKS.PROPERTIES.

- Ensure that the Service Manager is loaded when you run the backup tool.

To run the backup tool on NetWare or Windows NT/2000 servers,

1. At the inventory server console, enter **`StartSer DBBACKUP`**.

   View the status of the backup in the backup log file.

   The database will be copied to SYS:\ZENWORKS\DATABASE*directory_you_specify*.

To restore the database:

1. If the Inventory database server is up, stop the Storer service. At the database server console, enter **`StopSer Storer`**.
2. Exit the Sybase database.

   On NetWare servers, at the database server prompt, enter **`q`** to stop the Sybase database.

   On Windows NT/2000, stop the Sybase service (Adaptive Service Anywhere - ZENworks for Desktops 3).
3. Copy the backup files, overwriting the working database files.
4. Restart the database server.

The backup tool creates a log file, BACSTATUS.TXT, located in the ZENWORKS\DATABASE directory on NetWare and Windows NT/2000 servers. The log records the status of the backup operation. Open this text file to view the status of the backup. This file increases in size for every backup operation. Remove the existing contents of the file if you do not require the details.

Backing Up the Inventory Database (Oracle)

ZfD provides an option to back up the Inventory database from the server. However, if you have a database backup and restore method such as using Recovery Manager, we recommend that you continue to use your existing backup method.

To back up the database:

1. If the Inventory database server is up, stop the Storer service. At the database server console, enter **`StopSer Storer`**.

2 Load the Oracle Server Manager.

On NetWare server with Oracle 8i, enter **svrmgr31**.

On NetWare server with Oracle 8.0.4, enter **svrmgr30**.

On Windows NT/2000 server with Oracle 8i Enterprise Edition, from the taskbar, click Start > Run > enter **svrmgrl**.

3 Enter the following commands:

set instance *databaservername-databaseinstance-IPC*, where *databaseinstance* refers to the database instance that you have set up earlier. See "Configuring and Running Multiple Oracle Database Instances on a NetWare 5.x Server" on page 485.

For example, **set instance *austr-zfd3-ipc***.

4 Connect as an administrator. For example, if the administrator's internal name is *internal*, at the Server Manager prompt, enter **connect internal/*password***.

where *password* is the password created earlier. See "Configuring and Running Multiple Oracle Database Instances on a NetWare 5.x Server" on page 485.

4a At the Server Manager prompt, enter **select name from v$datafile;**

Displays the list of the datafiles that Workstation Inventory uses.

5 Ensure that no other databases are mounted. At the prompt, enter **shutdown normal**.

6 Disconnect and exit from the Server Manager. At the Server Manager prompt, enter **disconnect;**

Enter **exit;**

7 Copy the complete ZFD3\ORACLE directory to a backup volume or disk.

After the backup is done, ensure that the backup copy of the database matches the original copy. Perform database verification to verify the integrity of the backup.

To verify the database verification on a NetWare server with Oracle 8i, enter **load DBV81.NLM FILE=***path_to_the_database_file* **BLOCKSIZE=*4096***

To verify the database verification on a NetWare server with Oracle 8.0.4, enter **load DBV80.NLM FILE=*path_to_the_database_file* BLOCKSIZE=*4096***.

To verify the database verification on a Windows NT/2000 server with Oracle 8i, enter **DBV.EXE FILE=*path_to_the_database_file* BLOCKSIZE=*4096***

Example: enter **DBV.EXE FILE=*d:\zfd3\oracle\database\cim1.ora* BLOCKSIZE=*4096***

Also, run this command for the following files: CIM1.ORA, CIM2.ORA, CIM3.ORA, CIM4.ORA, CIM5.ORA, CIM6.ORA, CIM7.ORA, CIM8.ORA, CIM9.ORA, SYS1.ORA, and CTL1.ORA.

If the database backup is successful, ensure that there are no error messages on the verified pages. Ensure that the following displayed parameters display a zero value: TOTAL PAGES FAILING (DATA)=*0*, TOTAL PAGES FAILING (INDEX)=0, and TOTAL PAGES MARKED CORRUPT=0.

To restore the database:

1. If the Inventory database server is up, stop the Storer service. At the database server console, enter **StopSer Storer**.
2. Load the Oracle Server Manager.

   On a NetWare server with Oracle 8i, enter **svrmgr31**.

   On a NetWare server with Oracle 8.0.4, enter **svrmgr30**.

   On a Windows NT/2000 server with Oracle 8i Enterprise Edition, from the taskbar, click Start > Run > enter **svrmgrl**.
3. Connect as an administrator. For example, if the administrator's internal name is *internal*, at the Server Manager prompt, enter **connect internal/*password_for_administrator***.
4. Ensure that no other databases are mounted. Enter **shutdown normal**.
5. Disconnect and exit from the Server Manager. At the Server Manager prompt, enter **disconnect;**

   enter **exit;**
6. Copy the database from the backup location.

   If you copy the database to a different location than the earlier location, modify the location in the following files to specify the new path:

- Edit the INIT.ORA file located in \ZFD3\ORACLE\DATABASE to specify the new path for the following parameters:

  ```
  control_files=location_of_CTL1.ORA\CTL1.ORA

  background_dump_dest=location_of_TRACE_dir\TRACE

  user_dump_dest=location_of_TRACE_dir\TRACE
  ```

- Edit the _START.SQL file in the SYS:\SYSTEM to specify the location of INIT.ORA file in the following parameter:

  ```
  startup pfile=location_of_the_INIT.ORA\INIT.ORA
  ```

- Modify the location in the \ZFD3\ORACLE\ALTERCTRL.SQL to specify new path.

  For example, modify the existing DATA:\ZFD3\ORACLE\DATABASE path to ORACLE:\ZFD3\ORACLE\DATABASE in ALTERCTRL.SQL.

  In this .SQL file, modify the path for the following parameters, if required.

  ```
  startup nomount
  pfile=oracle:\zfd3\oracle\database\INIT.ORA

  logfile group 1
  'oracle:\zfd3\oracle\database\log1.ora' size 256K,

  logfile group 2
  'oracle:\zfd3\oracle\database\log2.ora' size 256K

  datafile 'oracle:\zfd3\oracle\database\sys1.ora',

  'oracle:\zfd3\oracle\database\rbs1.ora',

  'oracle:\zfd3\oracle\database\cim1.ora',

  'oracle:\zfd3\oracle\database\cim2.ora',

  'oracle:\zfd3\oracle\database\cim3.ora',

  'oracle:\zfd3\oracle\database\cim4.ora',

  'oracle:\zfd3\oracle\database\cim5.ora',

  'oracle:\zfd3\oracle\database\cim6.ora',

  'oracle:\zfd3\oracle\database\cim7.ora',

  'oracle:\zfd3\oracle\database\cim8.ora',
  ```

```
'oracle:\zfd3\oracle\database\cim9.ora',
'oracle:\zfd3\oracle\database\tmp1.ora'
```

Save the changes.

7 Load the restored database.

Removing the ZENworks 1.*x* Inventory Attributes from Workstations

If you are upgrading from ZENworks 1.*x* to ZfD3, the 1.*x* inventory attributes are replaced in the Directory Services schema by ZfD attributes during the installation. However, the 1.*x* attributes are not removed from Directory Services. You can use the Remove ZENworks 1.*x* Inventory Utility to remove these attributes from Directory Services. Though removal of the 1.*x* inventory attributes is not necessary to run inventory in ZfD, removing the attributes will free up space for other attributes and applications in Directory Services.

To remove the ZENworks 1.*x* attributes:

1 Run REMINV1X.EXE.

The REMINV1X.EXE file is found on the *ZENworks for Desktops* CD in the \PRODUCTS\ZENWORKS\CORE\CONSOLEONE\BIN directory.

2 Type the context for the workstations for which you want to purge the 1.*x* attributes.

3 Check the Include Subcontainers check box to include workstations located in the subcontainers of the selected context.

4 Click OK to run the utility.

A message box displays indicating that the attributes have been removed.

5 Click OK.

24 Using Workstation Inventory

This section contains the following information:

- “Viewing the Servers Deployed for Inventory” on page 511
- “Displaying Inventory Information” on page 511
- “Exporting the Inventory Data to CSV Format” on page 533

Viewing the Servers Deployed for Inventory

Using ConsoleOne™, you can view the servers and databases that you configured for collecting inventory.

To view the servers deployed for inventory:

1. In ConsoleOne, click a container > View > Complete Tree View.

   A Complete Tree View displays all the servers within the container.

2. In ConsoleOne, right-click an Inventory Service object > click View > click Up Tree View.

   A Up Tree View displays all the servers from the selected server up to the highest level (Root Server), including the database servers.

Displaying Inventory Information

The following sections will help you configure the Inventory database to display inventory information and explain the various types of information you can view:

- Listing of hardware and software components found on the managed workstation and any custom information you have specified for the workstation.

  The Workstation Inventory window displays the inventory items for a managed workstation. This window displays the data from the last inventory scan for the managed workstation.

  See “Viewing the Workstation Inventory Summary of a Managed Workstation” on page 512 for more information about viewing the inventory information of a managed workstation.

- Listing of managed workstations with the inventory information from the Inventory database satisfying the criteria you specify in the Inventory Query window. You form a query by specifying the component and its attribute for workstations within the selected database sites.

  See "Viewing Inventory Information of Managed Workstations Formed by Querying the Database" on page 525 for more information about querying the Inventory database.

- Listing of minimal information stored in the NDS Workstation object is available.

  See "Viewing the Minimal Inventory Information from an NDS Object" on page 527.

- Reporting Listing that generates the inventory information from the Inventory database specific to your needs.

  See Inventory Reports for more information about the types of reports.

Configuring the Inventory Database

If you want to view the inventory information of the database from ConsoleOne, you must configure the database. The inventory information from the Inventory database that you configure will be used for generating inventory reports, viewing inventory information, and for querying the inventory information from the database.

To configure the Inventory database:

1. In ConsoleOne, click a container.
2. Click Tools > Configure DB.
3. Select an existing ZENworks Database object from the list of Database objects.

   This Database object contains the database settings such as the protocol, port in use by the database, and others.
4. Click OK.

The database you configured is used for data retrieval unless you change it again using this same procedure.

Viewing the Workstation Inventory Summary of a Managed Workstation

ZfD3 provides the following inventory information collected from the managed workstations:

Scan Data Group	Scan Data Item	Description
HardwareSoftware Inventory > General	Asset Tag	Asset tag number that the ROM-based setup program creates
	Computer Model	Identifying information of the computer; for example; Compaq*, Dell*, and others
	Computer Type	Type of computer, such as IBM* PC, and others
	Machine Name	DNS name of the workstation
	ModelNumber	Serial number value for the computer, assigned during manufacture
	Primary Owner Contact	The phone number of the primary user of this system
	Primary Owner Name	The name of the primary user or owner of this system
HardwareSoftware Inventory > General > Login Details	Current login user	User logged in to the Primary NDS tree when the workstation was scanned
	Last login user	User logged in last to the Primary NDS tree when the workstation was scanned
Software > Software Vendors	Software Vendor Name	Name of the software manufacturer
	Name	Name of the software application
	Version	Version of the software

Scan Data Group	Scan Data Item	Description
Software > Operating System	Code Page	Language Code Page of the Operating System
	Description	Operating System description
	Install Date	Install date of the Operating System
	Name	Operating System name. For example, Windows* 95/ Windows 2000
	Size Stored in Paging Files	Page file size
	Version	Version of the Operating System
Software > Inventory Scan Information	Inventory Server	Name of the inventory server to which the scans are sent
	Version	Version of the Scanner
Hardware > Mouse Driver	Name	Name of the mouse driver
	Version	Version number of the mouse driver
Hardware > Modem	Caption	Modem label
	Description	Additional information about the modem
	Name	Identifying information of the modem
Hardware > Processor	Device ID	Special Hexadecimal string identifying the processor type
	Maximum Clock Speed (in MHz)	Maximum clock speed of the processor

Scan Data Group	Scan Data Item	Description
	Other Family Description	Additional description about the Processor Family, such as Pentium* Processor with MMX technology
	Processor Family	Identification of the processor family such as Pentium II, Pentium III, and others
	Processor Stepping	Single-byte code characteristic provided by microprocessor vendors to identify the processor model
	Role	Type of processor such as central processor, math coprocessor, and others
	Upgrade Method	The method by which this processor can be upgraded, if upgrades are supported
Hardware > BIOS	BIOS Identification Bytes	Byte in the BIOS that indicates the computer model
	Install Date	The manufacturing date of the BIOS
	Manufacturer	BIOS vendor name
	Name	BIOS label
	Primary BIOS	True state indicates Primary BIOS
	Serial Number	Serial number of the computer, assigned during manufacture
	Size	Size of the BIOS
	Version	Version or revision level of the BIOS
Hardware > Bus	Protocol Supported	Bus Protocol indicates PCI, ISA, and others

Scan Data Group	Scan Data Item	Description
	Version	Version of the Bus supported by the motherboard
Hardware > Keyboard	Delay	Delay before the repeat of a key
	Description	Description of the keyboard, such as IBM Enhanced 101 or 102 keys
	Layout	Layout of the keyboard
	Number of Function Keys	Total number of function keys
	Subtype	Type of the keyboard
	Typematic Rate	Rate of processing the keys
Hardware > Display	Current Bits Per Pixel	Number of adjacent color bits for each pixel
	Current Horizontal Resolution	Number of horizontal pixels shown by the display
	Current Vertical Resolution	Number of vertical pixels shown by the display
	Description	Description of monitor
	Max Memory Supported	Maximum memory that the display adapter supports for VIDEO RAM
	Max. Refresh Rate	Maximum refresh rate of the monitor for redrawing the display, measured in Hertz
	Min. Refresh Rate	Minimum refresh rate of the monitor for redrawing the display, measured in Hertz
	Number of Color Planes	Number of color planes supported by the video system

Scan Data Group	Scan Data Item	Description
	Video Architecture	The architecture of the video subsystem in this system, for example, CGA/VGA/SVGA/ 8514A
	Video Memory Type	The type of video memory for this adapter, for example, VRAM/SRAM/DRAM/EDO RAM
Hardware > Adapter	Description	Adapter description
Hardware > Display > Video Driver	Install Date	The Video BIOS release date
	Is Shadowed (True or false)	If true, the Video BIOS is currently being shadowed
	Manufacturer	Vendor name
	Version	The version number or version string of the Video BIOS
Hardware > Ports > Parallel Ports	Parallel Port Name	The logical name of the input-output device on this parallel port, under this operating environment
	DMA Support (True or False)	If true, DMA is supported
	Address	Base I/O address for this parallel port
	IRQ	IRQ number of the parallel port
Hardware > Ports > Serial Ports	Address	Base input-output address for this serial port
	IRQ Number	IRQ number of the serial port

Scan Data Group	Scan Data Item	Description
	Name	The logical name of the I/O device on this serial port, under this operating environment
Hardware > Disk > Floppy > Floppy Drive	Drive Letter	Mapped drive name of the floppy drive
Hardware > Disk > Floppy > Floppy Disk	Capacity	Floppy drive capacity
	Description	Floppy drive description
	Manufacturer	Vendor name
	Physical Cylinders	Floppy drive cylinders
	Physical Heads	Floppy drive R/W heads
	Sectors/Track	Floppy drive sectors per track
Hardware > Hard Disk	Capacity	Hard disk capacity
	Description	Description
	Manufacturer	Vendor name
	Physical Cylinders	Cylinders
	Physical Heads	Heads
	Sectors/Track	Hard disk drive sectors per track
Hardware > Hard Disk Logical Drive	Caption	Hard disk volume label
	Drive Letter	Letter name of the drive
	Size	Drive's actual size in MB
	Volume Serial Number	Hard disk volume serial number
Hardware > Battery	Chemistry	The battery chemistry, for example, Lithium-ion or Nickel metal hydride

Scan Data Group	Scan Data Item	Description
	Design Capacity	The design capacity of the battery in mWatt-hours
	Design Voltage	The design voltage of the battery in mVolts
	Install Date	The battery manufacture date
	Manufacturer	The name of the company that manufactured the battery
	Name	Device name for this battery, for example, Duracell* DR-36
	Serial Number	The serial number for this battery
	Smart Battery Version	The Smart Battery Data Specification version number supported by this battery
Hardware > Memory	Total Virtual Memory Size	Indicates the total number of bytes in the virtual address space of the calling process
	Physical Memory Size	Total Physical Memory of the workstation
Hardware > Sound Card > Sound Adapter	Description	Description of the multimedia component for the workstation
	Manufacturer	Vendor name
Hardware > Multimedia Card	Description	Description of the multimedia component for the workstation
	Manufacturer	Vendor name
	Name	Label of Multimedia card
Hardware > Backup Disk	Description	Description of the backup device for the workstation
	Name	Name of the backup device vendor

Scan Data Group	Scan Data Item	Description
Hardware > Backup Disk > Logical	Drive Letter	Name of the mapped drive to the backup device
Hardware > Inventory	Physical Memory (size in KB)	Total physical memory of the workstation
Hardware Software Inventory	Primary Owner Contact	Phone number of the primary user of this system
	Primary Owner Name	Name of the primary user or owner of this system
Hardware > Power Supply	Description	Expanded description of the input voltage capability for this power supply
	Total Output Power (in MilliWatts)	Attribute value that represents the total output power of the power supply
Hardware > Mouse	IRQ Number	Interrupt assigned to this device
	Name	Identifying information of the mouse
	Number of Buttons	Number of buttons on the mouse
Hardware > CDROM	Caption	CDROM Label
	Description	Description of the CDROM
	Manufacturer	Vendor Name
Hardware > CD ROM > Logical Drives	Drive Letter	Drive letter used for CD drives
Hardware > Local File System	File System Type	File system associated with this partition, for example, FAT/NFS
	Name	Name that the system uses to identify the partition, usually this is the drive letter

Scan Data Group	Scan Data Item	Description
	Size	Size of the partition
Hardware > Network Adapter Driver	Description	Description of the Network Adapter Driver
	Name	Name of the Network Adapter Driver
	Version	Version of the Network Adapter Driver
Hardware > Network Adapter	Max. Speed	Rate at which the data is transferred over the LAN
	Name	Network Adapter name
	Permanent Address	Node address stored permanently in the adapter
Network	IP Address	The unique address assigned to a computer on an IP Internet
	Subnet Mask	The subnet mask of the workstation paired with an IP address specifies to an IP router which octets or bits in the IP address are the network ID and which octets or bits are the node ID
	DNS Name	The DNS name of the workstation
	IPX Address	The IPX address of the workstation
	MAC Address	Unique node address permanently coded in the network adapter that identifies a specific computer on a network
Network	NetWare® Client™ Version	Version of the NetWare client software installed on the workstation

Scan Data Group	Scan Data Item	Description
System Information > System Cache	Associativity	Defines the system cache associativity (direct-mapped, 2-way, 4-way)
	Cache Type	Defines the system cache type, for example, Instruction, Data, Unified
	Capacity	Size of the data store where the cache information is kept
	Error Methodology	Error correction scheme supported by this cache component, for example, Parity/Single Bit ECC/MultiBit ECC
	Level	Indicates the cache level; internal cache that is built in to the microprocessors; external cache that is between the CPU and DRAM
	Line Size	Size in bytes of a single cache bucket or line
	Read Policy	Indicates whether the data cache is for read operation
	Replacement Policy	Algorithm that the cache uses to determine which cache lines or buckets should be reused
	Speed	Speed of this System Cache module in nanoseconds
	Write Policy	Indicates the two different ways (Write-Back and Write-Through Cache) that the cache can handle to write to the memory

Scan Data Group	Scan Data Item	Description
System Information > Motherboard	Number of Slots	The number of expansion slots in the motherboard for adding more memory, graphic capabilities, and support for special devices
	Version	Version of the motherboard
System Information > System IRQ	IRQ Number	Number of the Interrupt Request Line (IRQ), from 0 to 15
System > System DMA	Description	Name of the logical device that is currently using this DMA channel
	DMA Burst Mode	A data transmission mode in which data is sent faster than normal
	DMA Channel Number	Number of the Direct Memory Access (DMA) channel that a computer uses for transferring data to and from devices quicker than from computers without a DMA channel
	IRQ Availability	Indicates whether Virtual Direct Memory Access (DMA) is supported
System > Slot	Description	Card currently occupying this slot
	Max Data Width	Maximum bus width of cards accepted in the slot
	Thermal Rating	Maximum thermal dissipation of the slot in milliwatts

This window displays the data from the last inventory scan for the managed station.

For more information about the DMI data display, see “Viewing the DMI Data in the Inventory Summary Information” on page 523.

To view the inventory information of a managed workstation:

1. In ConsoleOne, click Tools > Configure DB.

   See “Configuring the Inventory Database” on page 512 for more information.

2. Right-click a managed workstation > click Actions > click Inventory.

Viewing the DMI Data in the Inventory Summary Information

The following list contains the DMI components that are scanned if they are instrumented and available in the DMI database on the workstation:

- BIOS Characteristics
- Disks
- DMA
- Electrical Current Probe
- IRQ
- Keyboard
- Network Adapter Driver
- Network Adapter Hardware
- Operating System
- Parallel Ports
- Pointing Device
- Power Supply
- Processor
- Serial Ports
- SubComponent Software
- System Cache
- System Contact Information
- System Enclosure
- System Hardware Security
- System Memory Settings

- System Power Controls
- System Resource DMA Info
- System Slot
- Temperature Probe
- Video
- Video BIOS
- Video Output Device
- Voltage Probe

See the DMTF Web site (http://www.dmtf.org) for more information on DMI.

Viewing Inventory Information of Managed Workstations Formed by Querying the Database

Using ConsoleOne, you query the Inventory database to display the hardware and software components of managed workstations that you want to view. The Inventory Query window displays the information satisfying the criteria you specify.

The Inventory database stores inventory data (hardware, memory, and environmental information) for each managed workstation. Querying the Inventory database helps to create groups of similar devices and to focus your reports on specific types of machines. For example, you can query the database to find machines that have an i486D processor and a VGA card.

HINT: If you are generating inventory reports from an Inventory database that has inventory information of more than 2,000 workstations, we recommend that you use the Data Export tool for better performance. The Data Export tool generates the inventory information in to a Comma Separated Value (.CSV) file. Use this .CSV file along with any standard reporting tool such as Seagate* Crystal Reports.

To query the Inventory database for inventory information:

1. In ConsoleOne, click a container.
2. Click Tools > Configure DB.

   See “Configuring the Inventory Database” on page 512 for more information.
3. Click Tools > Inventory Query.
4. Specify the criteria for query:

Find In: Choose a site name from the list. The query locates all workstations within the specified site satisfying the query expression.

Search Entire Database: Check this option to include all database sites while querying. The query locates all workstations in all sites satisfying the query expression.

Find Type: Select Quick or Advanced. Click Quick to specify a simple query. When you choose Quick query, you specify one attribute, relational operators, and the value of the attribute. Choose Advanced query to specify many attributes. Combine multiple query groups in which each group defines a set of query criteria. For example, use the Advanced query to run a query to discover all devices in the database with 486 processors and using Query connectors add another query to discover which of these workstations have a VGA color video adapter.

Attributes: Select the component attributes. Attributes that you can specify to query the managed workstations are as follows:

- Software (Name, Vendor, Version)
- Operating System (InstallDate, Name, Version, Code Page)
- Scanner Information (Version, LastScanDate, NumberofFilesScanned)
- NetWare Client details
- Hardware Information (Mouse, keyboard, video BIOS, display adapter, BIOS, processor, memory (virtual, physical), disk, floppy drives, hard disk, CD ROM drive, backup disk drives, ports (serial ports, parallel port), Bus, modem, sound card, battery, power supply, cache)
- Network (IP address, IPX™ address, Mac address, DNS)
- System (Motherboard, IRQ, Slot, DMA)

Operator: Form query groups that will be combined with the previous query group by using the relational operator specified between the query groups.

Value: Description values are the possible values of an inventory component. For example, 6.0 is a possible value for the DOS-Version attribute. Description values are not case-sensitive. Use the wildcard character % to substitute any number of characters, or the ? character to substitute one character in the Value field.

The list of description values displayed for an Inventory component is taken from the Inventory database corresponding to the component.

Save and Load the Existing Query Files: Save the queries in a file and load the query file as required.

5 Click Find.

This will query based on the query criteria you specify and display the workstations that match the query in the Query Results window.

In the Query Results window, double-click the workstation to view the inventory information of the workstation.

Viewing the Minimal Inventory Information from an NDS Object

The scanners store minimal scan data as an NDS Workstation object. You can view this minimal information from ConsoleOne. The Minimal Information page lists the inventory information of the scanned managed workstations.

See "Mapping Between Minimal Information Attributes and Attributes in the Inventory Database" on page 527 for more information about the listed items in this page.

To view the inventory information stored in NDS:

1 Right-click a managed workstation that has been successfully scanned > click Properties > click the ZENworks Inventory tab > click Minimal Information.

If you click More Workstation Information in this page, the Workstation Inventory window will be displayed.

Mapping Between Minimal Information Attributes and Attributes in the Inventory Database

The following table shows the mapping between minimal information attributes and attributes in the Inventory database:

Minimal Information	Inventory Database
Asset Tag	Scanned Inventory - Asset Tag
BIOS Type	Hardware - BIOS - Type
Computer Model	Scanned Inventory.Computer Model

Minimal Information	Inventory Database
Computer Type	Scanned Inventory.Computer Type
Disk Information	Hardware - Disk - Hard Disk - Drive Letter
IP Address	Network - IP Address
IPX Address	Network - IPX Address
Last Scan Date	Date and time when the workstation was scanned
MAC Address	Network - MAC Address
Memory Size	Hardware - Memory - Total Extended Memory
Model Number	Scanned Inventory.Model Number
NIC Type	Hardware - Network Adapter Driver - Description
Novell Client	Novell Client version
OS Type	Software - Operating Systems - OS - Name
OS.Version	Software - Operating Systems - OS - Version
Processor	Hardware - Processor
Serial Number	Workstation Serial Number
Subnet Mask	Network - Subnet Mask
Video Type	Hardware - Display - Type

Running Inventory Reports

You can run reports to gather inventory information from the Inventory database.

The inventory information is taken from the Inventory database you configure. For more information, see “Configuring the Inventory Database” on page 512.

Once you have configured the database, you access the hardware and software reports through the Tools menu from ConsoleOne. You can select from a predefined set of report forms to generate a report. The inventory report is displayed in the viewer window.

You can print or export the report as desired. Remember that any reports you generate will be empty if you haven’t configured ZfD to start populating the Inventory database with the data you want.

Before running the inventory reports, you must configure the inventory database. See “Configuring the Inventory Database” on page 512. The inventory reports always use the Inventory database you configured as the data source for your reports unless you change it later as described in “Configuring the Inventory Database” on page 512.

To generate the inventory report:

1. In ConsoleOne, click a server object.
2. Click Tools > Inventory Reports.
3. Click the report you want to generate.

   The description for the report is displayed on the right side of the screen.

   See Table 3, “Simple Inventory Lists,” on page 531 and Table 4, “Comprehensive Inventory Reports,” on page 532.
4. Specify the Selection Criteria.

   For example, if you want to view the inventory information of all database sites, select the report type you want and specify % as the Database Site Name. The report will display the inventory information of any database sites within the configured Inventory database. You can also specify the database site that you want.

   Depending on the type of report you want, you can filter the information. For example, to view all workstations of a particular database site with the Windows NT operating system, you select the Workstation Operating System Listing and specify the selection criteria Operating System Type as Windows NT, Operating System Version as 3.0, and the database site.

   See Table 2, “Selection Criteria for the Inventory Reports,” on page 529.
5. Click the Run Selected Report button.

A status box appears displaying the progress of the report generation. When the report is generated, it appears in the viewer. Use the buttons on the toolbar to page through, print, or export the report.

Table 2 Selection Criteria for the Inventory Reports

Report Name	Selection Criteria
Workstation Scan Time Listing	Database Site Name
Workstation Operating System Listing	Operating System Type, Operating System Version, Database Site Name
Workstation BIOS Listing	BIOS Install Date, Database Site Name
Workstation Processor Listing	Processor Family, Database Site Name
Workstation Processor Speed Listing	Lower Bound of Processor (in MHz), Upper Bound of Processor (in MHz), Database Site Name **HINT:** Ensure that the value for Lower Bound of Processor is less than the value for Upper Bound of Processor.
Workstation Video Adapter Listing	Video Architecture, Database Site Name
Workstation Network Adapter Listing	Network Adapter Name, Database Site Name
Workstation Software Listing	Software Name, Software Version, Database Site Name
Workstation Memory Listing	Lower Bound of Blocks of Memory (in MB), Upper Bound of Blocks of Memory (in MB), Database Site Name **HINT:** Ensure that the value for Lower Bound of Blocks of Memory is lesser than the value for Upper Bound of Blocks of Memory.
Software Summary Listing	Software Name, Software Version, Database Site Name
General Workstation Inventory Report	DN of the workstation, Database Site Name

Report Name	Selection Criteria
Asset Management Report	DN of the workstation, Database Site Name
Hardware Inventory Report	DN of the workstation, Database Site Name
Networking Information Report	DN of the workstation, Database Site Name
Software Inventory Report	DN of the workstation, Database Site Name
Software Inventory Report for the Entire Site	Database Site Name

If the Reporting dialog box allows wildcards, you can use an asterisk (*) and question mark (?), or their SQL equivalents, percent (%) and underscore (_). The wildcard characters can be used for character data only.

Example	Specifies to Include
%	All items
wNT%	All items starting with "wNT"
wNT_cpq	All items starting with "wNT" followed by any character and "cpq"
wNTcpq.xcorp	The single named item, in this case a workstation

You can generate the types of reports described below, assuming you have already configured ZfD to start populating the inventory database with the data you want. The following table lists the Simple Inventory lists that provide information on individual aspects of workstation inventory, such as operating system.

Table 3 Simple Inventory Lists

Report Name	Information Provided
Workstation Scan Time Listing	Date and time of the last inventory scan on each workstation
Workstation Operating System Listing	List of all the workstations with an OS type, an OS version, and the total number of such workstations
Workstation BIOS Listing	List of all the workstations with a BIOS release date, and the total number of such workstations
Workstation Processor Family Listing	List of all the workstations with a processor family (such as Pentium Pro), and the total number of such workstations
Workstation Processor Speed Listing	List of all the workstations within a range of processor speed (such as 200-400 MHz), and the total number of such workstations
Workstation Video Adapter Listing	List of all the workstations with a video adapter (such as MGA 2064W), and the total number of such workstations
Workstation Network Adapter Listing	List of all the workstations with a network adapter (such as 3Com* Fast EtherLink*) and the total number of such workstations
Workstation Software Listing	List of all the workstations with a software name, version, and the total number of such workstations
Workstation Memory Listing	List of all the workstations within a range of memory, and the total number of such workstations
Software Summary Listing	Lists of the total number of workstations with a particular software and version

The following table lists the Comprehensive Inventory Reports that combine several aspects of workstation inventory into each report, such as memory, hard disk, and processor.

Table 4 Comprehensive Inventory Reports

Report Name	Information Provided
General Workstation Inventory Report	BIOS, computer description, OS description, display details, NIC, physical disk drive, IP address, and MAC address for each workstation
Asset Management Report	BIOS, computer description, processor, and OS description for each workstation
Hardware Inventory Report	Memory, processor, display details, physical disk drive, and modem for each workstation
Networking Information Report	OS Description, MAC Address, NIC, and IP description for each workstation
Software Inventory Report	Software with product name, version, and vendor for each workstation
Software Inventory Report for the Entire Site	All softwares with product name, version, and vendor for the entire database site

To print a report:

1. Generate and view the report.
2. On the toolbar, click the printer icon.
3. In the Print dialog box, select the print options you want > click OK.
4. In the next Print dialog box, click OK to print in the default orientation mode.

   To change the mode, click Properties > click Orientation > select the mode (Landscape/Portrait) > click OK twice.

The Inventory Lists are designed to be printed in portrait mode; all the other reports are designed to be printed in landscape mode.

To export an inventory report to a file:

1. Generate and view the report.
2. On the toolbar, click the Export Report icon.
3. In the dialog box, specify the location and file format > click OK.

Exporting the Inventory Data to CSV Format

You can customize the inventory data you want to export from the ZfD3 Inventory database in to a comma separated value (CSV) file.

You select the inventory components that should be exported. For example, you choose the inventory component such as Operating System Name and Version. You can further filter the workstations whose attributes will be exported. For example, you can export only those workstations with a particular processor speed. The Data Export tool will export all workstations satisfying these query conditions in to a CSV file.

If required to reuse the same data export settings for export, you can save the data export configurations.

The following sections will help you use the Data Export tool:

- "Invoking the Data Export Tool" on page 534
- "Exporting the Inventory Data to a CSV File" on page 534
- "Forming the Query and Setting the Filter Conditions" on page 535
- "Loading Existing Configuration File" on page 537
- "Running the Data Export Program From the Server" on page 538

Invoking the Data Export Tool

To invoke the Data Export tool:

1. In ConsoleOne, click a container.
2. Click Tools > click Configure DB > select a ZENworks Database object. > click OK.
3. Click Tools > Data Export.

Exporting the Inventory Data to a CSV File

To export the inventory data to a CSV file:

1. Open the Data Export tool. See "Invoking the Data Export Tool" on page 534.
2. Select Create a New Database Query.

This option lets you add a new query that defines the inventory components such as hardware, software, network, and others that you want to export. You can also specify the criteria to limit the workstations and the database sites to be included in the query. Based on the inventory components and criteria you specify, the inventory data from the database is exported to a CSV file.

Click Next.

3. Select the database fields from the list of Database Fields > click Add button.

   If you select a group component, all subcomponents of the group are added. For example, if you select the Software component group, the subcomponents of Software such as vendor name, product name, and version are added.

   Click Next.

4. Specify the filter conditions for workstations, click Edit Query. See "Forming the Query and Setting the Filter Conditions" on page 535.

   Click Next

5. View the data export settings.

   Click Save Configuration to save the configurations settings to an .EXP file, specify the filename for the EXP file > click Save.

   The configuration file (.EXP) contains the settings such as the inventory components you selected, and also the query formed for filtering the workstation data export. You create an .EXP file so that you can reload the configuration settings and generate the .CSV files any time you need to.

   Click Next.

6. Click Perform the query from this Computer to run the data export processing from the workstation computer. This option will access the Inventory database on the specified database server and export the data in to a CSV file.

   To run the data export tool from a server, click Perform the Query on a Remote Server. See "Running the Data Export Program From the Server" on page 538.

   Click Finish.

7. Specify the .CSV filename > click OK.

This generates the .CSV file in the specified directory. Open the .CSV file in Microsoft* Excel, or any other CSV-supported viewer to view the exported data.

Forming the Query and Setting the Filter Conditions

To form the query and set the filter conditions for the data export:

1. In ConsoleOne, open the Data Export tool. See "Invoking the Data Export Tool" on page 534.
2. Select Create a New Database Query.
3. Set the following filter conditions.

   Selecting the database sites: In the Find In option, select the database sites names from the list.

   The Data Export tool will export the data from the Inventory database sites you select. This tool locates all workstations satisfying the filter conditions within the selected database sites.

   Click the Browse button to select one or more database site names from the list. In the Select Sites window, click one or more database sites. The selected site names are shown in double quotes.

   To include all database sites for data export, check the Search Entire Database checkbox. The data export tool will locate those workstations satisfying the filter conditions in all database sites.

   Selecting the attributes of the Inventory Components: Click the Browse Attribute button to select component attributes in the Select Attribute window. For example, to specify the version of BIOS as a component in the data export, select BIOS as the component, and select Version as the component attribute.

   Components are as follows: Software, Hardware, Network System, DMI, General Information.

   Operator: Relational operators show the relationship between the component and the value. Use the Matches option to specify the wildcard characters in the Value field.

   Specifying the values for the inventory attributes: Description values are the possible values of an inventory component. For example, 6.0 is a possible value for the DOS-Version attribute. Description values are not

case-sensitive. Use the wildcard character % to substitute any number of characters, or the ? character to substitute one character in the Value filed.

The list of description values displayed for an Inventory component is taken from the Inventory database corresponding to the component.

Specifying the query connectors and controls: The connectors and controls available for building filter conditions include the following:

AND: The expressions before and after the AND must be true.

OR: Either the expression before the OR or the expression after the OR must be true.

Insert Row: Lets you build the filter condition for this current row.

Delete Row: Deletes the row.

New Group: Lets you form a new filter condition group and specify the criteria for it. This group will be combined with the previous group by using the relational operator specified between the groups.

End: Ends the filter condition.

4 Click OK.

Loading Existing Configuration File

You can load an existing configuration file (.EXP). An .EXP file contains the settings such as the inventory components you selected, and also the query formed for filtering the workstation data export.

After you load the .EXP file, you can modify the settings for data export and then export the data to a CSV file.

To load existing configuration settings for data export:

1 Ensure that you have generated the data configuration files.

Complete the procedure outlined in “Exporting the Inventory Data to a CSV File” on page 534. This procedure generates the .CSV file and the data configuration files.

2 In ConsoleOne, open the Data Export tool. See“Invoking the Data Export Tool” on page 534.

3 Select Open a Saved Database Query > click Next.

The default directory for .EXP files is CONSOLEONE\version\REPORTING\EXPORT. Click Browse to open an existing .EXP file.

If the database setting specified in the .EXP file are not valid, the data export tool uses the default setting, which includes all databases for data export.

4 Click a saved database query from the list.

If you want to modify the existing query, click Edit. Otherwise, to proceed with the existing query, click Next.

5 View the data export settings. Click Next.

6 Click Perform the Query from this Computer to run the data export processing from the workstation computer. This option will access the Inventory database on the specified database server and export the data in to a CSV file.

To run the data export tool from a server, click Perform the Query on a Remote Server. See "Running the Data Export Program From the Server" on page 538.

Click Finish.

7 Specify the .CSV filename > click OK.

This generates the .CSV file in the specified directory. Open the .CSV file in Microsoft Excel, or any other CSV-supported viewer to view the exported data.

Running the Data Export Program From the Server

Running the Data Export program from a server is recommended if you are exporting data from a large database with more than 10,000 workstations or if you have specified complex queries with more than 20 database fields selected for exporting.

To run the data export program from the server:

1 Ensure that you have generated the data configurations files.

Follow the Step 1 to Step 5 outlined in "Exporting the Inventory Data to a CSV File" on page 534 and ensure that you save the settings in the .EXP file.

When you save an EXP file, a corresponding data configuration file is created in the same directory with the same filename as the .EXP file and with the .CFG file extension.

2. Click Perform the query on a Remote Server to run the data export program from any server that has Workstation Inventory components installed. Click Finish.

3. Copy the .EXP file and .CFG file to the server.

   These two files should exist in the same directory on the server.

   From the server console, run DBEXPORT.NCF on NetWare servers or DBEXPORT.BAT on Windows NT/2000 servers, type **DBEXPORT** *configuration_filename.EXP csv_filename.CSV*

   where *configuration_filename.EXP* is an existing file that contains the data export settings. The data exported from the database will be stored in the csv_filename.csv.

   The corresponding .CFG file for the .EXP file should be in the same folder as the .EXP file. The .CFG file contains the list of the database attributes to be exported.

4. Choose the Inventory sites.

   - To export the data from all database sites, satisfying the filter conditions, type 0.
   - To choose the database sites, type the numbers corresponding to the site names in the displayed list.
   - To select multiple site databases, separate the site numbers corresponding to the site names by commas.

   The data export displays the number of workstations that satisfy the query and filter conditions for export.

   Open the .CSV file in Microsoft Excel, or any other CSV-supported viewer to view the exported data.

25 Troubleshooting Workstation Inventory with Status Logs

ZENworks™ for Desktops lets you track whether the scan or the roll-up of information is successful by viewing the log files and scan history.

The inventory components report the status of the inventory scanning and roll-up of scan information in NDS®.

For example, viewing the scan logs, you can determine whether the scan was successful or if there were any errors while scanning the workstation or at the time of roll-up.

You can view the following status information:

- "Viewing the Scan History of a Workstation" on page 541
- "Viewing the Roll-Up History of the Server" on page 542
- "Viewing the Status of Inventory Components on a Server" on page 543
- "Viewing the Status of the Last Scanning in the Workstation Scan Log" on page 544
- "Viewing the Roll-Up Log of Servers" on page 545
- "Exporting the Inventory Status Log Files" on page 546
- "Quick Overview of Status Logs and Scan Logs" on page 546
- "Viewing the Status Log in XML Format" on page 547

Viewing the Scan History of a Workstation

The Scan Status reports the history of the scans done at the workstation. For example, you view the Scan Status window to determine whether the scan was successful for the workstation, or whether the Storer has stored the .STR files of the workstation in the database. The inventory components (Scanner and Storer) write the scan information in the Status log.

The Scan Status displays the history of the latest ten scans of the selected workstation.

The details of the log include:

Status Information	Details
Time of Scan	Displays the date and time of the scan.
Message	Displays the message reported by the inventory components while scanning the workstation and storing the scan data in the database. See Error Messages for more information about the error messages.

You can export the log file as a .CSV or TAB delimited file.

To invoke the Scan Status window:

1. In ConsoleOne™, right-click the workstation > click Properties > click the ZENworks Inventory tab > click Scan Status.

If the scan is disabled in the Inventory policy, the Scanner does not log any status messages in the Status Report log.

Viewing the Roll-Up History of the Server

The Roll-Up Status reports the status of the roll-up information from the server that initiated the roll up of data. For example, if your inventory setup consists of a Leaf Server which initiates the roll-up of data to the next-level Root Server, the Roll-Up log displays the roll-up history of the Leaf Server.

The inventory components of the server (Sender, Receiver, and Storer) write the scan information in the Roll-Up Status. For example, you view the Roll-Up log to determine whether there were any errors during roll-up of scan data from the server. This log also displays the most recent roll-up time of the scan data that was stored in the database on the topmost level server (Root Server). This log displays the history of the ten previous roll-up sessions done from the server.

The details of the log include:

Status Information	Details
Roll-Up Start Time	Displays the date and time of the roll-up.

Status Information	Details
Message	Displays the message reported by the inventory component while moving the scan data across the servers. See Error Messages for more information about the error messages.

You can export the file as a .CSV or TAB delimited file.

To invoke the Roll-Up Status window:

1. In ConsoleOne, right-click the Inventory Service object, from where the roll-up is done > click Properties > click Status Report tab > click Roll-Up Status.

Viewing the Status of Inventory Components on a Server

The Server Status window reports the status of the inventory server components on the selected server. You can view the Server Status log for any Inventory Service object. For example, you can determine whether the Sender sent the files to the Receiver or whether the Storer was able to establish the connection with the database successfully. The Server Status window displays the details of the ten latest status messages logged by the inventory server components.

If the inventory server components (Sender, Receiver, Selector, Storer, Service Manager, Roll-Up Scheduler) are not up and running on the server, the status of the server displays the information.

The details of the log include:

Status Information	Details
Time of Log	Displays the date and time when the message was reported by the inventory components.
Source	Displays the inventory component that has logged the status message.
Message Type	Displays the severity of the message.

Status Information	Details
Message	Displays the message reported by the inventory components. See Error Messages for more information about the error messages.

You can export the log file as a .CSV or TAB delimited file.

To view the Server Status window:

1. In ConsoleOne, right-click the Inventory Service object > click Properties > click Status Report > click Server Status.

Viewing the Status of the Last Scanning in the Workstation Scan Log

The Workstation Scan log reports the status of the latest scan done at the workstations associated with the selected Container. For example, you view the Workstation Scan log to determine whether the latest scans were successful for the workstations or whether the Storer has stored the workstation .STR files in the database. The inventory components (Scanner and Storer) write the scan information in the Status log.

You can also choose whether to display error, warning, and informational status messages of the workstations.

The details of the log include:

Status Information	Details
Scanned Workstation Name	Displays the DN of the workstation.
Time of Scan	Displays the date and time the status was logged.
Message Type	Displays the severity of the message.

Status Information	Details
Message	Displays the message reported by the inventory components while scanning the workstation or storing the inventory information in the database. See Error Messages for more information about the error messages.

You can export the file as a .CSV or TAB delimited file.

HINT: If the scan is disabled in the Inventory policy, the Scanner does not log any status messages in the Status Report log.

To view the Workstation Scan log window:

1. In ConsoleOne, click the container > Tools > Workstation Scan Log.
2. Click the message severity type > OK.

When the Scanner creates a .STR file of a workstation and the Selector processes this file, the inventory components also log the status of the scanning in the Workstation Status log. It is possible that at the same time, the Storer is processing the .STR files of another workstation. Meanwhile, the Scanner continues scanning and updates the Workstation Status log for the next scan. The Selector merges the .STR files of the previous scan and the current scan. In this scenario, the Storer processes the merged .STR files and updates the Workstation Scan log for the corresponding last scan.

Viewing the Roll-Up Log of Servers

The Roll-Up log reports the status of the latest roll-up from the Inventory Service objects in the Container. For example, you view the Roll-Up log to determine whether the latest roll up of information from the Roll-Up server for the Inventory Service object was successful. The inventory components (Sender, Receiver, and Storer) write the roll-up information in the Roll-Up log. You can also choose to display error, warning, and informational status messages of the Intermediate servers.

The details of the Roll-Up log include:

Status Information	Details
Roll-Up Initiated From	Displays the DN of the Intermediate server that initiated the roll-up.
Roll-Up Start Time	Displays the date and time the roll-up of information was initiated.
Source	Displays the inventory component that logs the status.
Message Type	Displays the severity of the message.
Message	Displays the message reported by the inventory components while scanning the workstation. See Error Messages for more information about the error messages.

You can export the log as a .CSV or TAB delimited file.

To invoke the Roll-Up Log window:

1. In ConsoleOne, click the container that contains the Inventory Service object > Tools > Roll-Up Log.
2. Click the severity type of the messages you want to view > OK.

Exporting the Inventory Status Log Files

You can store the details of the log files as Comma-Separated-Value reports or as a TAB delimited file.

To save the log as a file:

1. In ConsoleOne, open the Status window.
2. Click Export > choose the file type > type the filename > click OK.

Quick Overview of Status Logs and Scan Logs

The following table lists the status logs:

Log	Inventory Components that Log the Status	Details of the Log	How to View the Log File in ConsoleOne
Workstation Scan Log	Scan program, Storer	Scanned workstation name, time of scan, inventory component, message type, and status message	Click the container > Tools > Workstation Scan Log
Roll-Up Log	Sender, Receiver, Storer	Roll up initiated from, roll-up start time, inventory component, message type, status message	Click the container for the Inventory Service object > Tools > Roll-Up Log

The following table lists the scan logs:

Scan Status	Inventory Components that log the Status	Details of Log	How to View the Log file in ConsoleOne
Workstation Scan Status	Scan program, Storer	Time of scan and status message	In ConsoleOne, right-click the workstation > click Properties > click the ZENworks Inventory tab > click Scan Status
Status of Inventory components on Server	Sender, Receiver, Selector, Storer, Service Manager, Roll-Up Scheduler	Time of log, source, message type, message	In ConsoleOne, right-click the Inventory Service object > click Properties > click Status Report > click Server Status
Roll-Up Status	Sender, Receiver, Storer	Roll up start time, message	In ConsoleOne, right-click the Inventory Service object > click Properties > click Status Report > click Roll-Up Status

Viewing the Status Log in XML Format

All inventory components log the status messages in a log file maintained in XML (Extensible Markup Language) format. Unlike the status logs that contain a history of the ten latest status messages, the status XML log stores all status messages.

The log file contains the following data:

- Inventory module name
- Date and time of status logging
- Severity of the message
- Message text and status message number
- DN name, if the inventory module is associated with a particular DN object in NDS
- Product specific details of the module

The format of the log file is as follows:

```
?xml version="1.0" encoding="UTF-8"?>

?xml stylesheet type="text/xsl" href="inventorylog.xsl"?

<message_log>

 <message_entry>

   <module_name>Scanner</module_name>

   <severity>Critical</severity>

   <date_time>8/3/00 12:49 PM</date_time>

    <message_tag>unable to create scan data files</
message_tag>

    <dn_name>Inv_server</dn_name>

  </message_entry>

  </module_name>Storer</module_name>

    <severity>Critical</severity>

    <date_time>8/3/00 12:49 PM</date_time>

    <message_tag>unable to update the database</message_tag>

    <dn_name>Inv_server</dn_name>

</message_entry>

..

</message_log>
```

A sample stylesheet and Document Type Declaration (DTD) file are located in SYS:\PUBLIC\ZENWORKS on the server.

The INVENTORYLOG.XML log file is located in the SYS:\PUBLIC\ZENWORKS\WMINV\LOGS directory on NetWare® and Windows* NT/2000 servers.

By default, the maximum size of the log file is 100 KB. To modify the maximum size of the log file, edit the INVENTORYLOG.INI file. On NetWare and Windows NT/2000 servers, this file is in the SYS:\PUBLIC\ZENWORKS directory.

The contents of INVENTORYLOG.INI is as follows:

```
max_file_size=100 KB
```

Modify the MAX_FILE_SIZE parameter, if required.

If the filesize exceeds the value specified in the MAX_FILE_SIZE parameter, the file is archived as *filename*_OLD.XML. The latest messages will be in the current log file.

To view the log data file, use a third-party XML browser.

If you want to view the XML log file in languages other than English, see Viewing the Inventory Status Reports (XML format) in languages other than English.